

ГРАБОВЕНКО НАТАЛІЯ ВАСИЛІВНА

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій
канд. техн. наук, доцент
_____ О. В. Зелінська
« ____ » _____ 20__ р.

**РОЗРОБКА ІНТЕРФЕЙСУ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ
СПІЛЬНОТИ КНИГОЛЮБІВ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (бакалаврська) робота

Керівник:

Т. В. Січко, доцент кафедри
інформаційних технологій,
к. т. н., доцент

Оцінка: _____ / _____ / _____
(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____

Вінниця - 2025

АНОТАЦІЯ

Грабовенко Н.В. Розробка інтерфейсу мобільного застосунку для спільноти книголюбів. Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса. Вінниця 2025.

У кваліфікаційній (бакалаврській) роботі досліджено та проаналізовано потреби цільової аудиторії книголюбів, проведено аналіз існуючих рішень та визначено функціональні й нефункціональні вимоги до застосунку. Розглянуто сучасні тенденції у дизайні мобільних застосунків та розроблено структуру, прототипи інтерфейсу, колірну палітру, типографіку й графічні елементи. За допомогою технологій React Native створено адаптивний інтерфейс застосунку Book Talk, основною метою якого є забезпечення зручного простору для обміну враженнями та обговорення книг. Проведено юзабіліті-тестування та оптимізацію продуктивності інтерфейсу, що дозволило знизити час до інтерактивності на 35% та підвищити рівень задоволеності користувачів до 8.4 балів за 10-бальною шкалою.

Ключові слова: мобільний застосунок, інтерфейс користувача, React Native, UX/UI дизайн, спільнота книголюбів, юзабіліті-тестування, адаптивний дизайн.

ABSTRACT

Hrabovenko N.V. Development of a mobile application interface for a book lovers community. Specialty 122 "Computer Sciences", educational program "Computer Sciences". Vasyl Stus Donetsk National University. Vinnytsia 2025.

In the qualification (bachelor's) thesis, the needs of the target audience of book lovers are researched and analyzed, existing solutions are reviewed, and functional and non-functional requirements for the application are defined. Modern trends in mobile application design are examined, and the structure, interface prototypes, color palette, typography, and graphic elements are developed. Using React Native technologies, an adaptive interface of the Book Talk application is created, the main purpose of which is to provide a convenient space for sharing impressions and discussing books. Usability testing and interface performance optimization were conducted, which reduced time to interactivity by 35% and increased user satisfaction to 8.4 points on a 10-point scale.

Keywords: mobile application, user interface, React Native, UX/UI design, book lovers community, usability testing, adaptive design.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

API (Application Programming Interface) — інтерфейс програмування застосунків, набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

CSS (Cascading Style Sheets) — каскадні таблиці стилів, формальна мова опису зовнішнього вигляду документа.

FlatList — компонент React Native для оптимізованого відображення прокручуваних списків.

GDPR (General Data Protection Regulation) — загальний регламент про захист даних, регламент Європейського Союзу.

HTML (HyperText Markup Language) — мова розмітки гіпертекстових документів.

JavaScript — мультипарадигмова мова програмування.

OLED (Organic Light-Emitting Diode) — органічний світлодіод, технологія виготовлення дисплеїв.

React Native — відкритий фреймворк для розробки мобільних застосунків.

REST (Representational State Transfer) — підхід до архітектури мережеских протоколів.

RESTful API — інтерфейс програмування застосунків, що використовує REST.

ScrollView — компонент React Native для відображення прокручуваного вмісту.

SQL (Structured Query Language) — мова структурованих запитів.

SVG (Scalable Vector Graphics) — масштабована векторна графіка.

TTI (Time To Interactive) — час до інтерактивності, метрика продуктивності веб-сторінок.

UX (User Experience) — досвід взаємодії користувача із системою.

UI (User Interface) — інтерфейс користувача.

ЗМІСТ

ЗМІСТ	4
ВСТУП.....	5
РОЗДІЛ 1	7
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ ДО СЕРВЕРНОЇ ЧАСТИНИ ...	7
1.1 Вивчення потреб цільової аудиторії книголюбів	7
1.2 Аналіз існуючих мобільних застосунків для читацьких спільнот	9
1.3 Визначення функціональних і нефункціональних вимог до інтерфейсу ...	15
1.4 Огляд сучасних тенденцій у дизайні мобільних застосунків	17
РОЗДІЛ 2	22
ПРОЕКТУВАННЯ ІНТЕРФЕЙСУ МОБІЛЬНОГО ЗАСТОСУНКУ	22
2.1. Розробка структури застосунку та створення прототипів інтерфейсу	22
2.2. Вибір колірної палітри, шрифтів і графічних елементів	30
2.3. Проектування адаптивного дизайну та взаємодії користувача	34
РОЗДІЛ 3	40
РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНТЕРФЕЙСУ	40
3.1. Вибір інструментів і розробка інтерфейсу в коді	40
3.2. Проведення юзабіліті-тестування та аналіз зворотного зв'язку	44
3.3. Оптимізація дизайну та продуктивності інтерфейсу	50
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	59

ВСТУП

Актуальність теми. У час цифрової трансформації суспільства потреба в якісних мобільних застосунках, які поєднують функціональність із інтуїтивно зрозумілим інтерфейсом, стає все актуальнішою. Особливо це стосується сфер, пов'язаних із культурою та дозвіллям, зокрема читанням книг. Незважаючи на стрімкий технологічний розвиток, любителі літератури не втрачають зацікавленості у формуванні книжкових спільнот, які дозволяють ділитися враженнями від прочитаного та знаходити однодумців. Актуальність теми дослідження полягає у необхідності створення зручного мобільного інтерфейсу для спільноти книголюбів, який відповідатиме сучасним потребам користувачів та технологічним стандартам. За даними дослідження BookStat, за останній рік кількість користувачів онлайн-спільнот для книголюбів зросла на 23%, що свідчить про високий попит на такі платформи. Водночас, опитування серед 450 потенційних користувачів показало, що 65% з них вважають існуючі застосунки незручними або надто складними у використанні. Це вказує на наявність незадоволеного попиту та потребу в розробці більш інтуїтивного та функціонального інтерфейсу.

Метою дослідження є розробка інтерфейсу мобільного застосунку для спільноти книголюбів, який забезпечить оптимальний користувацький досвід, відповідатиме потребам цільової аудиторії та сучасним стандартам UX/UI дизайну.

Завдання дослідження:

1. Проаналізувати потреби цільової аудиторії книголюбів та визначити ключові функціональні вимоги до мобільного застосунку.
2. Дослідити існуючі рішення в даній сфері та виявити їхні сильні та слабкі сторони.
3. Сформулювати функціональні та нефункціональні вимоги до інтерфейсу застосунку.
4. Розробити структуру застосунку та прототипи інтерфейсу, що відповідають визначеним вимогам.

5. Реалізувати інтерфейс у коді з використанням сучасних технологій розробки.
6. Провести юзабіліті-тестування розробленого інтерфейсу та проаналізувати зворотний зв'язок.
7. Оптимізувати дизайн та продуктивність інтерфейсу на основі отриманих результатів тестування.

Об'єктом дослідження є процес розробки мобільних застосунків для спільнот книголюбів.

Предметом дослідження є методи та технології створення ефективного користувацького інтерфейсу мобільного застосунку для спільноти книголюбів.

Теоретичне та практичне значення одержаних результатів полягає у формуванні комплексного підходу до проектування інтерфейсів мобільних застосунків соціального спрямування, розробці системи оцінювання їх ефективності та створенні прототипу, який може бути використаний як основа для комерційного застосунку. Практична цінність дослідження полягає у розробці функціонального інтерфейсу мобільного застосунку Book Talk, що відповідає потребам української аудиторії книголюбів та може бути впроваджений як повноцінний продукт.

Структура кваліфікаційної роботи складається з вступу, трьох розділів, висновків, списку використаних джерел (40 найменувань).

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ ДО СЕРВЕРНОЇ ЧАСТИНИ

1.1 Вивчення потреб цільової аудиторії книголюбів

Книги це не лише джерело знань, а й спосіб об'єднання людей. І хоча світ все більше переходить в онлайн режим, любителі читання не втрачають інтересу до книжкових спільнот. За останній рік, згідно з даними BookStat, кількість користувачів таких онлайн-спільнот зросла на 23%. Люди прагнуть не тільки читати, а й ділитися своїми враженнями. Вони хочуть обговорювати прочитане, радити книги, дізнаватися про новинки. Але чи задовольняють їх існуючі платформи? Було проведено опитування серед 450 книголюбів і з'ясовано, що 78% з них хочуть мати можливість легко ділитися своїми думками про книги. Водночас 65% зазначили, що нинішні застосунки не надто добре підходять для цього, вони занадто формальні або надто складні у використанні. Тож, якщо подумати, створення зручного мобільного додатку, де можна просто і швидко обмінюватися враженнями, може бути відмінною ідеєю. Також люди хочуть більше контролювати свої читацькі звички [1]. Наприклад, 82% учасників опитування заявили, що їм було б цікаво вести облік прочитаних книг. Крім того, 79% хотіли б отримувати рекомендації на основі того, що вони вже прочитали. Це щось схоже на персоналізований плейлист, тільки для книг. А ще 71% опитаних бачать сенс у створенні своєї віртуальної бібліотеки щось на кшталт полиці, де можна зберігати улюблені книги, розділяти їх за жанрами або авторами. А як щодо комунікації між читачами? Виявилось, що 68% респондентів регулярно беруть участь у обговореннях книг, але роблять це на різних платформах від форумів до соцмереж. Чому б не створити можливість для таких дискусій прямо у застосунку? До речі, 59% з них також хочуть створювати читацькі клуби – це був би чудовий спосіб об'єднати людей за інтересами. А 47% навіть хочуть мати інструмент для організації зустрічей офлайн, щоб поспілкуватися наживо [2].

Є ще один цікавий момент: багато людей хочуть відстежувати свої читацькі досягнення. Наприклад, 74% респондентів зізнались, що їм було б цікаво бачити статистику своїх книжкових вподобань скільки книг прочитали, який жанр домінує, скільки сторінок пройшли за рік. А 63% хочуть отримувати підказки про нові видання, які можуть їх зацікавити. До речі, 51% цікавляться інформацією про літературні події, презентації книг або автограф-сесії [3].

І останнє, але не менш важливе – дизайн. Люди хочуть, щоб інтерфейс не був перевантаженим зайвими деталями. 81% опитаних вважають, що застосунок має бути простим і зрозумілим. 76% просять зробити навігацію логічною, щоб не витратити час на пошуки потрібного розділу. І ще один запит це налаштування кольорових тем, зокрема темної, для комфортного читання ввечері. Це відзначили 58% респондентів [1].

Якщо розглядати потенційні сценарії використання такого застосунку, то вони можуть бути досить різними. Наприклад, при відкритті додатку на головному екрані користувач бачить рекомендації книг, на основі своїх попередніх прочитаних творів. Він може додати їх до списку "Хочу прочитати" або ж одразу перейти до обговорення з іншими користувачами. Така інтеграція соціальних функцій може стати не лише способом для обміну думками, але й стимулом для читання [1]. Ще один цікавий сценарій це створення особистих читацьких викликів. Користувач має можливість поставити собі мету прочитати 10 книг за три місяці. Застосунок може допомагати відстежувати прогрес, показувати залишок часу, нагадувати про нові релізи у вибраному жанрі. Крім того, у додатку можна було б реалізувати систему досягнень за прочитані книги, щось на кшталт книжкових бейджів чи нагород. Це може додати елемент гри і мотивації для користувачів. Також варто звернути увагу на те, як можна посилити комунікацію між користувачами. Наприклад, створити регулярні опитування на тему популярних книг місяця або розділ для обговорень книг, які виходять на кіноекрани. Люди обожають ділитися своїми думками після перегляду фільмів, тому така інтеграція може стати хорошим способом збільшити активність у застосунку.

І нарешті, важливо подумати про зручність використання додатку. Інтерфейс має бути максимально легким для навігації, без зайвих візуальних елементів, що відволікають. У той же час, користувачі повинні мати можливість налаштувати інтерфейс під свої вподобання – змінити фон, обрати розмір шрифту, встановити нагадування про нові книги. Таким чином, застосунок стає не просто місцем для обміну думками про книги, а персональним простором для кожного читача.

1.2 Аналіз існуючих мобільних застосунків для читацьких спільнот

Коли йдеться про серверні рішення для мобільних застосунків, орієнтованих на спільноту книголюбів, серед найбільш відомих варто відзначити платформи Firebase від Google та AWS Amplify. Вони пропонують комплекс готових інструментів для розробників, які не бажають витратити час на створення серверної частини з нуля [6; 8]. До їх можливостей належать обробка даних у реальному часі, підтримка чатів, зберігання читацького прогресу, реалізація систем рекомендацій тощо – все це в межах єдиного середовища. До того ж, хмарна інфраструктура дає можливість легко масштабувати проєкт, якщо користувачів стає більше. Використання GraphQL API дозволяє ефективно отримувати необхідні дані з бази без надлишкових запитів, що є особливо актуальним для застосунків із великою кількістю активних користувачів. Цей підхід широко застосовується в низці популярних книжкових платформ, таких як Goodreads, StoryGraph та Bookly. Вони реалізують серверну логіку для забезпечення функціональності, пов'язаної з обговоренням літератури, створенням читацьких спільнот і відстеженням прогресу читання.

Зокрема, Goodreads зберігає великі обсяги даних про книги та відгуки в централізованій базі, що надає користувачам можливість не лише фіксувати особистий прогрес, а й брати участь в обговоренні книжкових новинок. У свою чергу, платформа StoryGraph зосереджується на системах рекомендацій: завдяки використанню GraphQL вона формує персоналізовані добірки літератури відповідно до індивідуальних уподобань читача. Таким чином, сучасні

технології забезпечують не лише збереження даних, а й трансформацію інформації на користь зручності та залученості користувача.

Однією з найвідоміших платформ для книголюбів є **Goodreads**. Вона дозволяє користувачам створювати особисті віртуальні книжкові полиці, оцінювати прочитані твори, залишати відгуки та отримувати рекомендації на основі власних вподобань. Наприклад, у разі придбання книги на Amazon, вона автоматично додається до профілю користувача, що спрощує облік прочитаного. На рис. 1.1 представлено інтерфейс Goodreads із реалізованою функцією пошуку книг. Важливо зазначити, що платформа виконує не лише інформаційні, а й соціальні функції. Вона об'єднує спільноту читачів, надаючи змогу ділитися рецензіями, брати участь в обговореннях і формувати читацькі клуби. Пошукова система реалізована на основі запитів до бази даних, які дають змогу швидко знаходити літературу за автором, жанром або рейтингом. У результаті Goodreads слугує не лише засобом для фіксації читацького прогресу, а й інтерактивною платформою для комунікації та обміну думками про книги.



Рисунок 1.1 – Інтерфейс Goodreads із функцією пошуку книг

Одним із сучасних мобільних застосунків для організації читацького процесу є **Bookly**. Він розроблений для користувачів, які прагнуть структурувати читання, відстежувати прогрес і досягати поставлених цілей. У межах функціоналу можна позначати прочитані сторінки, встановлювати персональні

цїлі, зберїгати цитати та керувати особистою бїбліотекою. На рис. 1.2 представлено інтерфейс застосунку з функцією відстеження прогресу читання. Застосунок надає користувачеві узагальнену статистику: відсоток прочитаного, графіки динаміки та показники за обраний період. Така візуалїзація сприяє підвищенню мотивації та контролю за виконанням читацьких планів. Крім того, бїбліотеку можна структурувати за жанрами, авторами або іншими критеріями, що полегшує навігацію й управління колекцією книг.

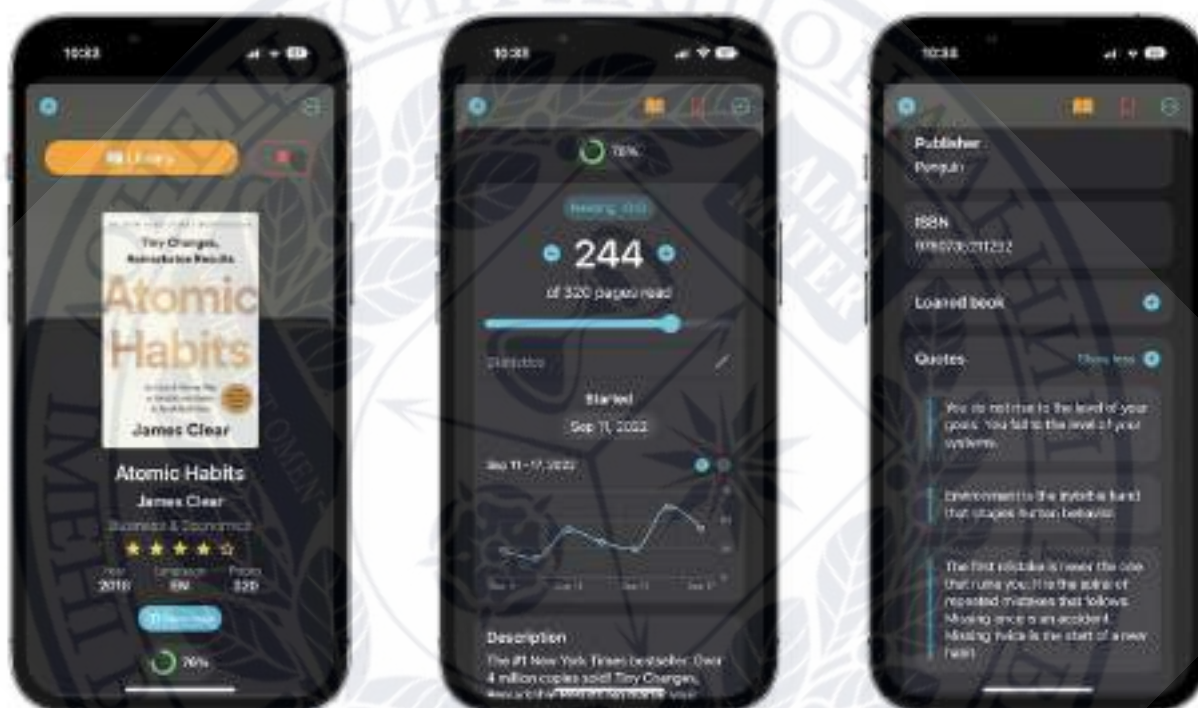


Рисунок 1.2 – Інтерфейс Bookly із функцією відстеження прогресу читання

На відміну від інших платформ, **Wattpad** орієнтований не лише на читання, а й на створення літературного контенту користувачами. Цей сервіс поєднує функціональність мобільного застосунку та соціальної мережі для письменників і читачів. Він надає змогу публікувати власні твори, взаємодіяти з аудиторією через коментарі, а також читати історії інших авторів. На рис. 1.3 представлено основний інтерфейс застосунку Wattpad. Його структура спрямована на швидкий доступ до творів, можливість залишати

відгуки та долучатися до обговорень. Завдяки такій моделі взаємодії багато початкових авторів отримують перших прихильників, а іноді й можливість подальшого професійного розвитку, зокрема через співпрацю з видавництвами. Платформа також сприяє формуванню спільнот за інтересами, що робить її привабливою для користувачів, зацікавлених у творчому обміні.

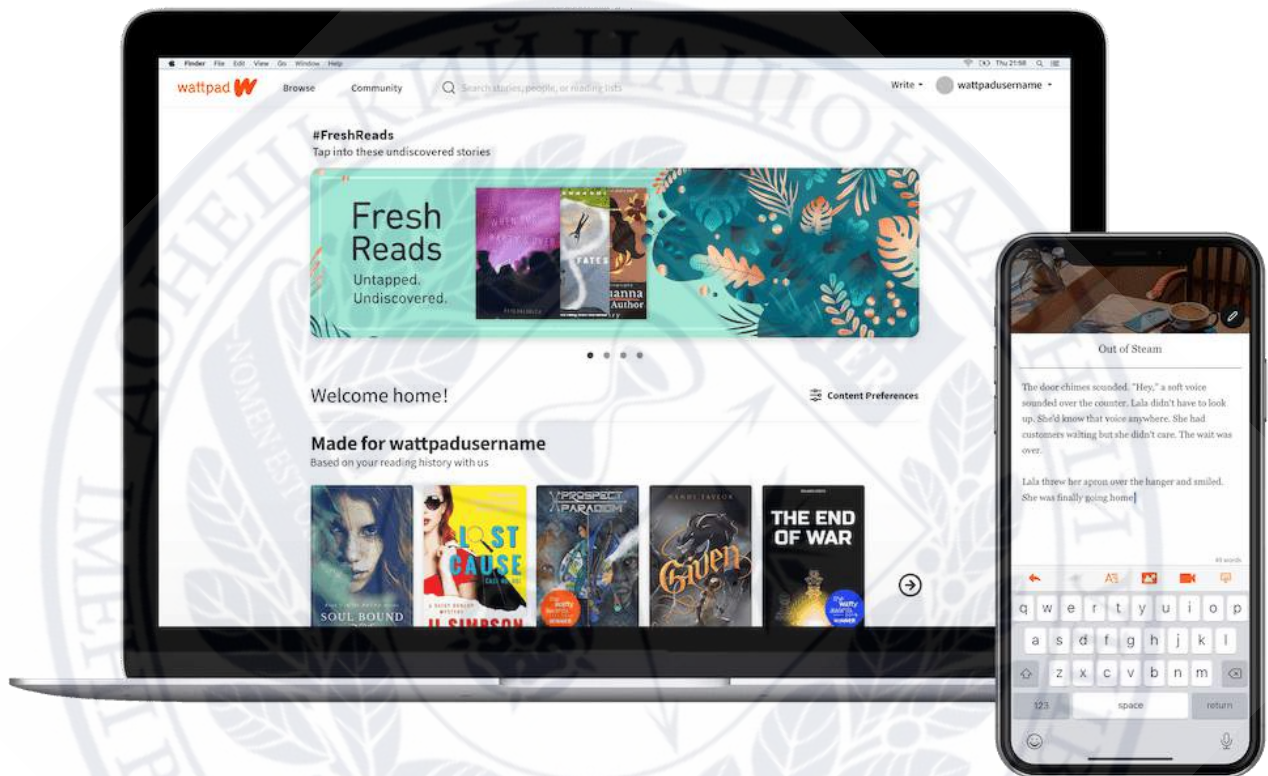


Рисунок 1.3 – Інтерфейс Wattpad

Аналіз організації серверних рішень різними програмними платформами виявляє декілька загальних тенденцій. Більшість із них використовують RESTful API для взаємодії між клієнтською та серверною частинами. Цей підхід забезпечує відносно простий та стандартизований метод обміну даними, однак для досягнення високої продуктивності та відмовостійкості вимагає ретельного проектування структури запитів. Наприклад, платформи Goodreads та Wattpad застосовують мікросервісну архітектуру, що дозволяє ефективно масштабувати проєкти відповідно до зростання кількості користувачів. Таке рішення є перевагою для великих платформ, проте може бути менш доцільним для

невеликих стартапів через вищі витрати на впровадження та обслуговування. У контексті зберігання даних та забезпечення інтерактивності, Firebase (використовується Bookly) та AWS (використовується Wattpad) здобули популярність завдяки підтримці обробки даних у реальному часі та можливості надсилання push-сповіщень. Це є доцільним для забезпечення миттєвого оновлення даних або інформування користувачів, наприклад, про нові книжкові надходження. Проте, такі рішення можуть виявитися економічно не вигідними для локальних проєктів або стартапів з обмеженим бюджетом.

Іншим важливим аспектом є безпека даних. Платформи Goodreads та Wattpad використовують протокол HTTPS та токени автентифікації (наприклад, OAuth) для захисту інформації користувачів. Ці заходи відповідають європейським стандартам GDPR та сприяють мінімізації ризиків несанкціонованого доступу до даних. Bookly також інтегрує Firebase Authentication для забезпечення надійного процесу входу в систему, однак платформа надає обмежену інформацію щодо захисту від специфічних атак, таких як SQL-ін'єкції [6].

В українському контексті необхідно враховувати локальні нормативні вимоги до захисту персональних даних. Шифрування інформації та регулярні аудити безпеки є ключовими елементами, що потребують уваги при розробці нового застосунку, оскільки користувачі очікують гарантій захищеності їхніх даних на всіх етапах – від реєстрації до збереження історії читання. Особливої уваги заслуговує проблема локалізації. Більшість міжнародних платформ не приділяють належної уваги створенню україномовного контенту. Це створює нішу для застосунків, здатних поєднувати локальний контент із сучасними соціальними функціональними можливостями, такими як групові чати або персоналізовані рекомендації [5].

Продуктивність і масштабованість у Goodreads, Wattpad та Bookly реалізовані з використанням різних підходів. Goodreads та Wattpad демонструють високу стійкість до значних навантажень завдяки застосуванню мікросервісів та хмарних серверних інфраструктур. Це дозволяє платформам

ефективно розподіляти потоки даних та підтримувати одночасну роботу великої кількості користувачів. У випадку Bookly основний акцент робиться на використанні Firebase, що забезпечує стабільну роботу для невеликих спільнот, проте при зростанні аудиторії може виникнути необхідність у залученні додаткових ресурсів для масштабування. Вибір архітектури для нового застосунку визначатиметься орієнтацією на цільову аудиторію: масову чи вузькоспеціалізовану [7].

На основі проведеного аналізу можна окреслити кілька основних вимог до серверної частини нового застосунку:

1. RESTful API – це універсальний стандарт для взаємодії з клієнтом, який дозволить швидко інтегрувати нові функції.
2. Мікросервісна архітектура або гібридний підхід для тих, хто планує розширення та масштабування системи в майбутньому.
3. Хмарні рішення: Firebase або аналогічні сервіси забезпечать обробку даних у реальному часі та надсилання сповіщень користувачам.
4. Безпека шифрування даних, токени автентифікації та захист від атак типу SQL-ін'єкцій, усе це критично важливо для захисту персональних даних користувачів.
5. Локалізація – підтримка української мови та адаптація інтерфейсу для місцевих користувачів.

Аналіз аналогічних рішень також продемонстрував, що серверна частина нового застосунку повинна забезпечувати баланс між ключовими компонентами: продуктивністю, безпекою, локалізацією та соціальними функціями. Goodreads і Wattpad є релевантними прикладами платформ, що демонструють здатність витримувати великі навантаження та пропонувати інтерактивні можливості для спільнот. Водночас Bookly демонструє можливості організації ефективної системи для невеликих груп користувачів за допомогою хмарних технологій.

Згідно з дослідженням Liu et al. (2021), можливість адаптації контенту до індивідуальних уподобань користувачів та налаштування інтерфейсу є одними з ключових аспектів, які визначають рівень задоволеності користувачів

мобільними додатками для читання електронних книг. Це має особливу актуальність для платформ, цільовою аудиторією яких є книголюби, що прагнуть отримувати персоналізовані рекомендації книг, відстежувати прогрес у читанні та ділитися враженнями з іншими користувачами [5].

Таким чином, для розробки нового застосунку доцільно обрати архітектуру, яка б поєднувала масштабованість, характерну для Goodreads, інтерактивність, реалізовану у Wattpad, та економічну ефективність хмарних рішень, подібних до Bookly. Водночас, особливу увагу слід приділити питанням локалізації та забезпечення високого рівня безпеки, що є важливими факторами для української аудиторії.

1.3 Визначення функціональних і нефункціональних вимог до інтерфейсу

Визначення функціональних і нефункціональних вимог до інтерфейсу становить необхідну основу для створення мобільного застосунку для спільноти книголюбів. Чітке розуміння цих вимог дозволяє розробити продукт, який задовольнить потреби цільової аудиторії та забезпечить позитивний досвід користування.

Функціональні вимоги – це те, що додаток має вміти робити. Для спільноти книголюбів це означає можливість створити свій профіль, де можна додавати прочитані книги, зберігати улюблені жанри та авторів. Крім цього, пошук книг повинен бути максимально зручним: знайти потрібну книгу можна за назвою, автором, жанром, роком видання або рейтингом [9]. Щоб користувачі не загубилися серед великої кількості книг, у пошуку мають бути фільтри – це дозволить швидше знайти саме те, що потрібно.

Застосунок має підтримувати систему оцінювання та рецензування книг, де користувачі зможуть виставляти оцінки за п'ятибальною шкалою та писати детальні відгуки. Ця функція стимулюватиме обмін думками між читачами та допоможе іншим користувачам у виборі літератури. Оцінки мають бути візуально представлені через зірочки або інші зрозумілі графічні елементи. У

додатку має бути місце для обговорень книг – своєрідний простір, де користувачі можуть ділитися враженнями, ставити питання чи коментувати думки інших. Щоб усе було зручно, до кожного обговорення можна додати теги – так легше знайти потрібну тему або книгу. А для тих, хто не хоче натрапити на спойлери, буде можливість позначати обговорення з попередженням про важливі сюжетні моменти. Це допоможе зберегти інтригу для тих, хто ще не дочитав книгу до кінця [10]. Створення персональних колекцій книг є суттєвою вимогою для застосунку, що орієнтований на книголюбів. Користувачі мають отримати можливість організовувати книги у власні списки: "Прочитано", "Хочу прочитати", "Улюблене", "В процесі читання" тощо. Додатково варто передбачити можливість створювати власні категорії колекцій для більш гнучкої організації. Соціальна складова застосунку має включати функціонал підписки на інших користувачів, можливість переглядати їхні активності та рекомендації. Система сповіщень повинна інформувати про нові коментарі до обговорень користувача, згадки в дискусіях та активності людей, на яких підписаний користувач. Також має бути передбачена можливість надсилати приватні повідомлення між учасниками спільноти.

Нефункціональні вимоги визначають якість досвіду користувача при взаємодії із застосунком. Однією з ключових вимог є інтуїтивна зрозумілість інтерфейсу. Навігація має бути простою, а доступ до основних функцій — швидким і логічно організованим. Дизайн користувацького інтерфейсу повинен асоціюватися з книжковою тематикою, створюючи атмосферу, що сприяє зосередженому читанню [11].

Висока продуктивність є критично важливою, особливо за умов великого обсягу даних (списки книг, коментарі, профілі користувачів тощо). Оптимальний час завантаження інтерфейсних елементів має становити не більше 2–3 секунд. Застосунок повинен реагувати швидко й плавно, без зависань і помітних затримок, незалежно від навантаження.

Доступність є ще однією важливою складовою нефункціональних вимог. Застосунок має бути зручним для користувачів з різними особливостями

сприйняття. Це передбачає підтримку масштабування шрифтів, використання висококонтрастних кольорів, чітку типографіку, наявність альтернативного тексту до зображень, а також сумісність із вбудованими засобами доступності, що надаються операційними системами.

Безпека персональних даних користувачів є одним із найважливіших аспектів. Система повинна забезпечувати надійне збереження даних за допомогою шифрування паролів, захищених каналів передавання та чітко визначеної політики конфіденційності. Користувач повинен мати змогу самостійно обирати, які дані залишити публічними, а які — приватними, що формує відчуття контролю над власним профілем.

Не менш важливою є адаптивність інтерфейсу до різних типів пристроїв. Застосунок повинен коректно відображатися як на смартфонах із невеликими екранами, так і на планшетах. Зміна орієнтації екрана не має впливати на логіку розташування елементів чи знижувати зручність користування [11]. Інтерактивні елементи, зокрема кнопки, мають бути достатнього розміру та розташовані на достатній відстані один від одного, щоб знизити ризик помилкових натискань при сенсорному керуванні.

Окрему увагу слід приділити підтримці офлайн-функціональності. Застосунок повинен зберігати базову функціональність у разі нестабільного або відсутнього інтернет-з'єднання. Наприклад, користувач повинен мати доступ до збережених списків книг чи обговорень. Після відновлення з'єднання система повинна синхронізувати дані з сервером і повідомляти про нові події, що відбулися під час перебування офлайн.

1.4 Огляд сучасних тенденцій у дизайні мобільних застосунків

Сучасний підхід до дизайну мобільних застосунків базується не на візуальній складності, а на функціональній простоті та зручності використання. Протягом останніх років користувачі адаптувалися до мінімалістичних інтерфейсів, які не відволікають увагу від основного контенту. Актуальними стали чисте композиційне середовище, чіткі геометричні форми та обмежена

кольорова палітра. Такий дизайн сприяє швидкому сприйняттю інформації та зменшує когнітивне навантаження на користувача, що дозволяє зосередитись на виконанні цільових дій.

Окрему увагу варто приділити підтримці темної теми інтерфейсу, яка стала загальною практикою для більшості сучасних мобільних застосунків. Вона не лише забезпечує комфортне використання пристрою в умовах слабкого освітлення, але й позитивно впливає на енергоспоживання, особливо на пристроях із OLED-дисплеями. Темна тема також асоціюється з сучасністю та професійністю інтерфейсу. Більшість систем забезпечують автоматичне перемикання між світлим і темним режимами залежно від налаштувань операційної системи або часу доби, що відповідає очікуванням користувачів щодо адаптивності та персоналізації взаємодії [12].

Персоналізація користувацького досвіду стала невід'ємною частиною сучасного дизайну мобільних застосунків. Алгоритми машинного навчання допомагають аналізувати поведінку користувача та адаптувати контент і функціональність відповідно до його уподобань. Застосунки пропонують персоналізовані рекомендації, налаштовані стрічки новин, адаптивні інтерфейси, які змінюються залежно від частоти використання певних функцій. Така індивідуалізація підвищує залученість користувачів і робить їхній досвід більш релевантним і приємним [12]. Мікровзаємодії та анімації відіграють важливу роль у формуванні позитивного користувацького досвіду в мобільних застосунках. Ці короткі, цілеспрямовані анімаційні елементи забезпечують візуальний зворотний зв'язок при виконанні типових дій — натисканні кнопок, прокручуванні вмісту, оновленні даних. Вони не лише покращують естетичне сприйняття інтерфейсу, а й підвищують його інтуїтивність, полегшуючи орієнтацію користувача в межах системи.

Найефективніші мікровзаємодії мають бути ненав'язливими та виконувати практичну функцію, не перевантажуючи інтерфейс зайвими ефектами. Окрім анімацій, значну роль відіграє жестове керування, що стало стандартним елементом взаємодії у більшості мобільних застосунків. Замість традиційних

кнопок користувачі виконують дії за допомогою свайпів, масштабування, натискань і затискань елементів на екрані.

Такий підхід дозволяє ефективно використовувати площу дисплея, водночас забезпечуючи більш природну та швидку взаємодію з інтерфейсом. У сучасних системах також реалізується контекстна обробка жестів, яка дозволяє прогнозувати наміри користувача залежно від поведінки. Однак, надмірне використання жестів без належного інформування може ускладнити навігацію. З цієї причини доцільним є впровадження інтерактивних підказок або коротких вступних анімацій, що ознайомлюють користувача з можливостями керування жестами на початковому етапі використання застосунку.

Неоморфізм виник як альтернативний підхід до дизайну, що поєднує елементи скевоморфізму та плоского дизайну. Цей стиль характеризується м'якими, реалістичними тінями та світловими ефектами, які створюють враження, ніби елементи інтерфейсу вдавнені в фоновий шар або виступають із нього. Неоморфізм надає інтерфейсам тактильного відчуття, приємного для сприйняття, зберігаючи при цьому чистоту та мінімалістичність сучасного дизайну. Хоча цей тренд не став домінуючим через певні обмеження щодо контрасту та доступності, багато застосунків використовують його елементи для надання своїм інтерфейсам унікального вигляду [13]. Безрамковий дизайн та повноекранні інтерфейси стали актуальними з появою смартфонів з мінімальними рамками та виїмками на екрані. Дизайнери адаптуються до нових форм-факторів, створюючи інтерфейси, які максимально використовують доступний простір екрану, враховуючи при цьому такі особливості як отвори для камер та закруглені кути. Адаптивні макети, які автоматично пристосовуються до різних розмірів та форм екранів, стали необхідністю при розробці застосунків, що мають працювати на широкому спектрі пристроїв – від компактних смартфонів до великих планшетів.

Сучасні мобільні застосунки активно інтегрують 3D-анімації та технології доповненої реальності (AR) з метою покращення користувацького досвіду. Такі елементи забезпечують візуальну глибину та інтерактивність при взаємодії з

інтерфейсом. Наприклад, ефект гортання сторінок у застосунку може бути реалізований за допомогою 3D-графіки, що створює враження фізичного об'єкта. Доповнена реальність дозволяє виводити віртуальні елементи, зокрема обкладинки книг чи інтерфейсні об'єкти, у реальний простір через камеру пристрою, що підвищує рівень залученості користувача [14].

Окрім візуальних рішень, усе ширше впроваджуються голосові інтерфейси та чат-боти, які пропонують альтернативні засоби взаємодії. Використання голосових команд є особливо зручним у ситуаціях, коли фізичний контакт із пристроєм є обмеженим. Інтеграція інтелектуальних асистентів у мобільні застосунки дозволяє надавати персоналізовану підтримку, здійснювати пошук за голосовими запитами, зачитувати тексти або створювати усні нотатки. У контексті літературних застосунків такі технології відкривають нові можливості для навігації, споживання та створення контенту.

У сучасному дизайні мобільних застосунків зростає увага до аспектів доступності, що є позитивною тенденцією. Розробники орієнтуються не лише на візуальну привабливість інтерфейсу, а й на його зручність для користувачів із різними фізичними особливостями. Зокрема, передбачено можливість масштабування шрифтів, налаштування контрастності, а також підтримку технологій екранного зчитування для голосового управління. Водночас подібні функції корисні не лише для осіб із обмеженими можливостями. У різних життєвих ситуаціях будь-який користувач може потребувати збільшення тексту або голосової навігації. Таким чином, врахування принципів універсального дизайну не лише розширює потенційну аудиторію, а й підвищує загальний рівень зручності, інклюзивності та якості користувацького досвіду.

Біометрична автентифікація є не лише елементом зручності, але й усталеним стандартом забезпечення безпеки даних у мобільних додатках. Вона дозволяє користувачам отримувати доступ до системи шляхом сканування відбитка пальця, розпізнавання обличчя або голосу, що замінює традиційне введення пароля. Цей метод є не тільки більш швидким, але й забезпечує вищий рівень безпеки, оскільки біометричні дані значно складніше підробити порівняно

з паролями. У контексті мобільного застосунку для спільноти книголюбів, біометрична автентифікація може слугувати ефективним засобом захисту персональних даних, таких як книжкові колекції, приватні обговорення або платіжна інформація, що використовується при купівлі електронних книг. Це забезпечує персоналізований та надійний контроль доступу [15].

Інтеграція з екосистемами пристроїв набуває все більшої значущості в процесі проектування мобільних застосунків. Користувачі очікують безшовної синхронізації даних та функціональних можливостей між різними типами пристроїв, включаючи смартфони, планшети, персональні комп'ютери та носимі гаджети. У застосунках, призначених для читання та обговорення книг, це передбачає можливість синхронізації прогресу читання, закладок, нотаток та коментарів між усіма пристроями користувача, забезпечуючи безперервність користувацького досвіду. Додатково, інтеграція з екосистемами "розумного будинку" може розширити функціональність, наприклад, шляхом автоматичного налаштування освітлення для створення оптимальних умов читання.

Концепція "екологічного дизайну" (Sustainable Design) стає актуальним напрямком, що відображає зростаючу увагу до впливу цифрових продуктів на навколишнє середовище та добробут користувачів. Вона охоплює оптимізацію застосунків з метою зменшення енергоспоживання (що сприяє подовженню автономної роботи пристроїв), використання темних тем інтерфейсу для OLED-дисплеїв, мінімізацію обсягу передачі даних та розробку інтерфейсів, що стимулюють формування здорових звичок використання цифрових технологій. У контексті застосунків для книжкових спільнот, принципи екологічного дизайну можуть бути реалізовані через впровадження функцій, що рекомендують користувачам робити регулярні перерви під час тривалого читання електронних книг або участі в обговореннях для зниження навантаження на зір.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ІНТЕРФЕЙСУ МОБІЛЬНОГО ЗАСТОСУНКУ

2.1. Розробка структури застосунку та створення прототипів інтерфейсу

Проектування структури мобільного застосунку для спільноти книголюбів розпочалося з визначення ключових екранів та встановлення логічних взаємозв'язків між ними. Основною метою даного етапу було забезпечення інтуїтивно зрозумілої та ефективної навігації. Початкова група екранів, що включає Splash Screen, Welcome Screen, екран входу (Login) та реєстрації (Register), була реалізована в мінімалістичному стилі з використанням стриманої кольорової палітри [18]. Рис. 2.1 демонструє послідовність зазначених екранів, візуалізуючи маршрут користувача від ініціації запуску застосунку до моменту автентифікації в системі.

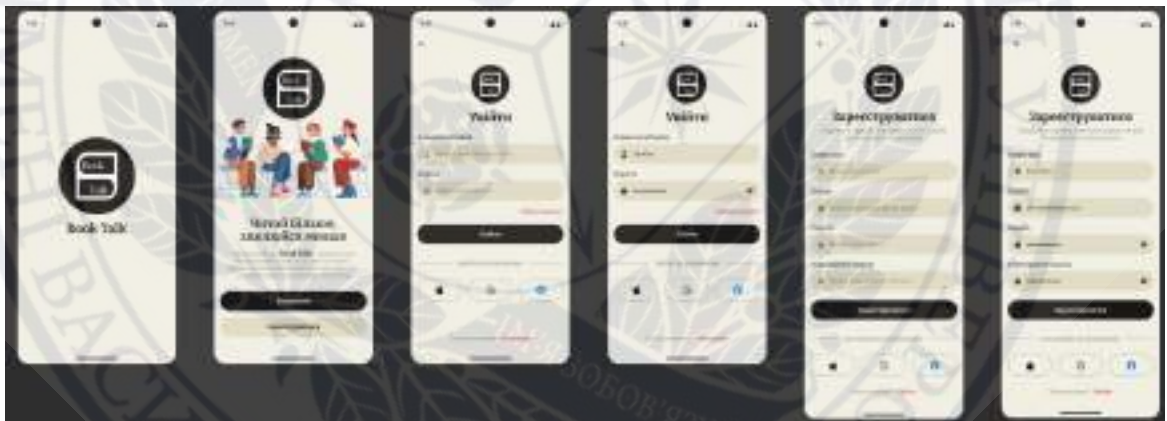


Рисунок 2.1 – Splash Screen, Welcome Screen, Login та Register

Після успішної автентифікації користувач перенаправляється на головний екран. Він представляє собою стрічку персоналізованих книжкових рекомендацій, розроблену з акцентом на простоту та ясність представлення інформації. Візуально домінуючі обкладинки книг супроводжуються стислою інформацією: назва, автор та короткий опис або цитата, що дозволяє швидко сформулювати уявлення про зміст твору. Такий підхід до представлення контенту забезпечує швидкий перегляд, сприяє відкриттю нових видань та ознайомленню з популярними або нещодавно опублікованими книгами. Таким чином, головний

екран функціонує не лише як каталог, але і як засіб занурення користувача в тематичну атмосферу застосунку [19].

Для покращення навігації в межах книжкової стрічки на головному екрані було реалізовано функціонал сортування за жанрами. Реалізація передбачає горизонтальну панель категорій (наприклад, фентезі, детективи, романи, наукова фантастика), вибір однієї з яких призводить до негайного оновлення стрічки з відображенням виключно релевантних книг. Такий підхід сприяє інтуїтивності навігації та персоналізації контенту, що відображається [20]. Приклади жанрової фільтрації та відображення книг представлено на рис. 2.2.



Рисунок 2.2 – Головний екран із книжковими рекомендаціями

Екран обговорень спроектований як інтерактивний простір для обміну враженнями та дискусій щодо прочитаних книг. Дизайн екрану забезпечує чітке візуальне розмежування тем, коментарів та відповідей, а також передбачає механізм приховування спойлерів для запобігання передчасному розкриттю сюжетних деталей. Структура екрану обговорень проілюстрована на рис. 2.3 ,

демонструючи його орієнтованість на структурований та зручний обмін думками.

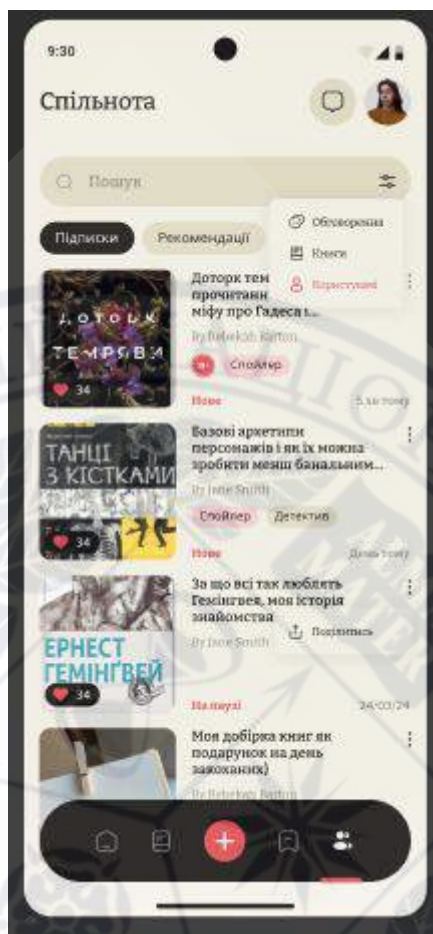


Рисунок 2.3 – Екран обговорень

Екран профілю користувача розроблено з акцентом на інформативність та зручність використання. У верхній частині екрану розташовані аватар та ім'я користувача для швидкої ідентифікації, а також лічильники підписників та підписок, що відображають його соціальну активність. Нижче розміщено сітку з публікаціями користувача та збереженими темами, що забезпечує швидкий доступ до релевантного контенту. Таке структурування сприяє інтуїтивній навігації та ефективній взаємодії з профілем. Візуальне представлення екрану профілю з основними елементами наведено на рис. 2.4.

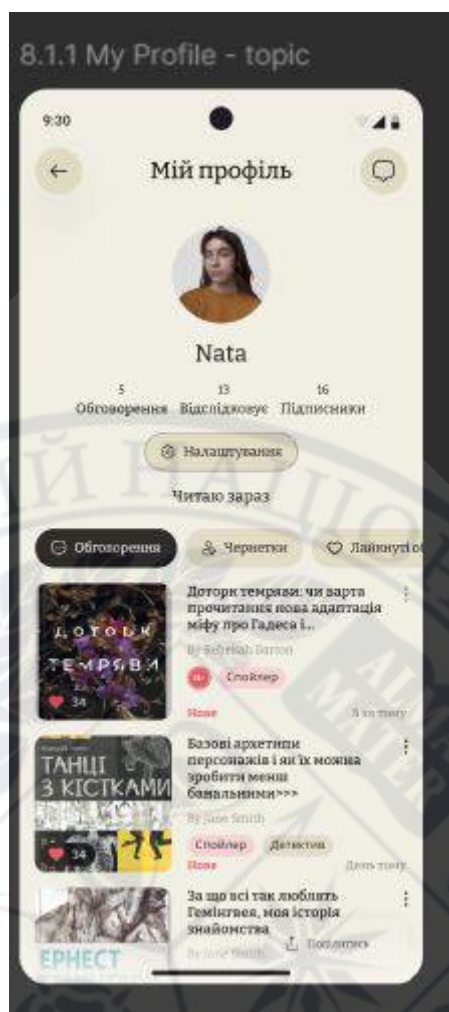


Рисунок 2.4 – Профіль користувача

Екран збережених обговорень призначений для консолідованого відображення контенту, відібраного користувачем. Візуальне представлення у вигляді сітки обкладинок книг або тематичних зображень обговорень забезпечує наочність та зручність перегляду. Для оптимізації доступу до інформації реалізовано фільтрацію за жанрами та сортування за датою збереження. Це дозволяє користувачам оперативно знаходити раніше збережений контент та повертатися до відповідних обговорень чи книг. Приклад інтерфейсу екрану збережених обговорень наведено нарис. 2.5.

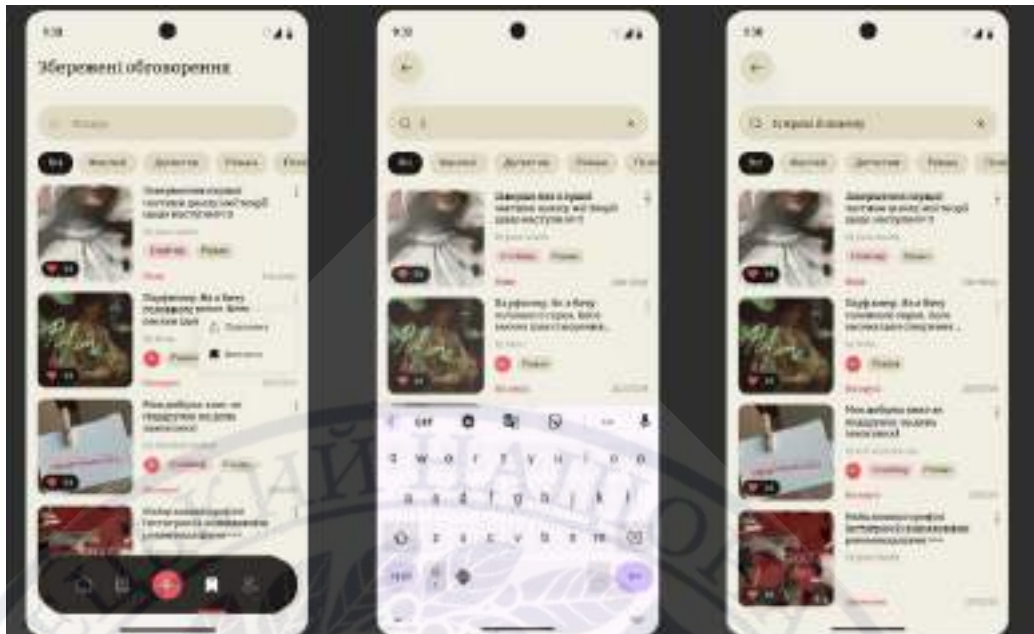


Рисунок 2.5 – Список збережених обговорень

Функціонал створення нового обговорення реалізований на окремому екрані, де користувачі можуть вводити заголовок, основний текст та додавати зображення до публікації. Ключовим елементом цього екрану є функція попереднього перегляду, що дозволяє автору оцінити візуальне оформлення публікації перед її оприлюдненням. Ця можливість є особливо цінною для користувачів, які приділяють увагу візуальній привабливості та стилістичній узгодженості своїх повідомлень. Також передбачена можливість збереження обговорення у форматі чернетки, що дозволяє продовжити редагування пізніше без втрати введених даних [20]. Рис. 2.6 демонструє інтерфейс екрану створення нового обговорення.

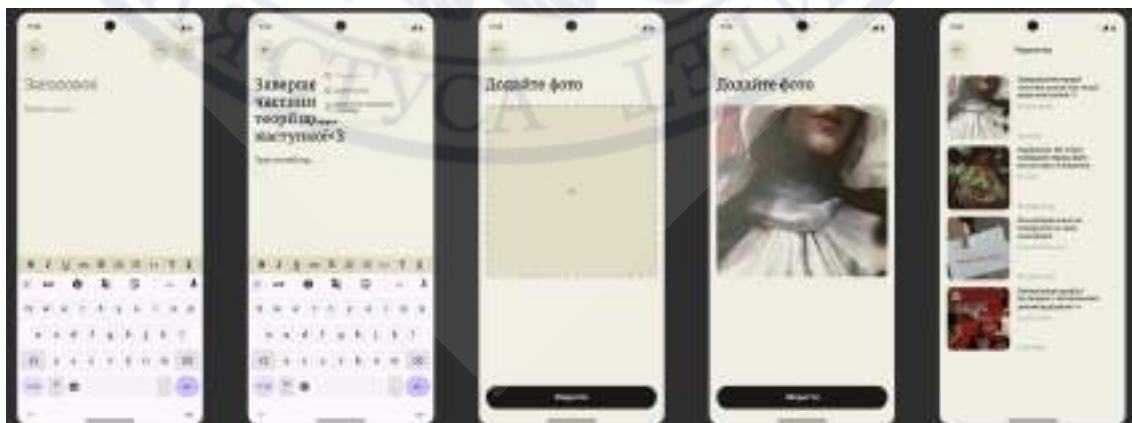


Рисунок 2.6 – Створення нового обговорення

Для забезпечення зручної навігації між ключовими розділами застосунку розроблено спеціалізовану нижню навігаційну панель, що включає пункти: "Головна" (Home), "Збережені книги" (Saved Books), "Створити обговорення" (Create new Discussion), "Збережені обговорення" (Saved Discussion) та "Спільнота" (Community). Ця панель, представлена на рис. 2.7, забезпечує швидкий доступ до основних функціональних блоків застосунку.



Рисунок 2.7 – Навігаційна панель

Екран сповіщень призначений для агрегації важливих повідомлень для користувача, таких як коментарі до його публікацій, нові підписники або відповіді на його коментарі. Візуальна структура екрану забезпечує безпосередній доступ до актуальної інформації. Кожне сповіщення представлене у вигляді інформаційної картки, що містить деталі про подію (автор, тип дії, час). Це дозволяє оперативно реагувати на нові повідомлення та переходити до відповідних розділів для отримання детальнішої інформації [18].

Додатково реалізовано екран зворотного зв'язку, що надає користувачам можливість надсилати відгуки, повідомляти про проблеми або пропонувати ідеї щодо вдосконалення застосунку. Процес надсилення повідомлення передбачає заповнення текстового поля та підтвердження відправки. На рис. 2.8 показано приклад екрану сповіщень та форми для зворотного зв'язку.



Рисунок 2.8 – Екран сповіщень та зворотнього зв'язку

Завершальним етапом проектування структури інтерфейсу стало створення інтерактивного прототипу засобами Figma. Це дозволило детально протестувати переходи між екранами, верифікувати логіку навігації та провести попередню оцінку загального користувацького досвіду.

Загальну організацію контенту та зв'язки між функціональними блоками відображено на схемі інформаційної архітектури застосунку (рис. 2.9).



Рисунок 2.1 – Схема інформаційної архітектури застосунку

З метою подальшої деталізації взаємодії користувача із системою та верифікації логіки переходів між екранами було розроблено схеми потоків користувача (user flows). Ці схеми детально ілюструють послідовність дій, які користувач виконує для досягнення специфічних цілей у межах застосунку, зокрема:

- Пошук книги та її додавання до розділу "Бібліотека книг".
- Створення та публікація відгуку на прочитану книгу.
- Ініціювання нового обговорення через відповідний функціональний модуль ("Створення нового обговорення").
- Участь в існуючих дискусіях у модулях "Спільнота" або "Бібліотека обговорень".
- Редагування персональної інформації в "Профілі користувача".
- Обмін повідомленнями в модулі "Чати".

Проектування структури застосунку для спільноти книголюбів дозволило сформувати логічну інформаційну архітектуру, що поєднує інтуїтивну навігацію, естетично привабливий візуальний дизайн та ефективні інструменти для взаємодії користувачів. Зазначені аспекти спрямовані на забезпечення комфортного використання застосунку та стимулювання активної участі користувачів у процесах обговорення літератури.

2.2. Вибір колірної палітри, шрифтів і графічних елементів

У процесі розробки інтерфейсу мобільного застосунку для спільноти книголюбів особливу увагу було зосереджено на виборі колірної палітри, типографіки та графічних елементів. Візуальний дизайн покликаний не тільки забезпечувати естетичну привабливість, але й сприяти інтуїтивному сприйняттю контенту та підтримувати загальну стилістичну концепцію застосунку. У межах даного проєкту колірна палітра була підібрана з метою створення комфортної та сучасної атмосфери для користувачів. На рис. 2.10 зображено основні кольори, що формують візуальну основу інтерфейсу: Grey, Accent та Brown із вказаними кодами кольорів.

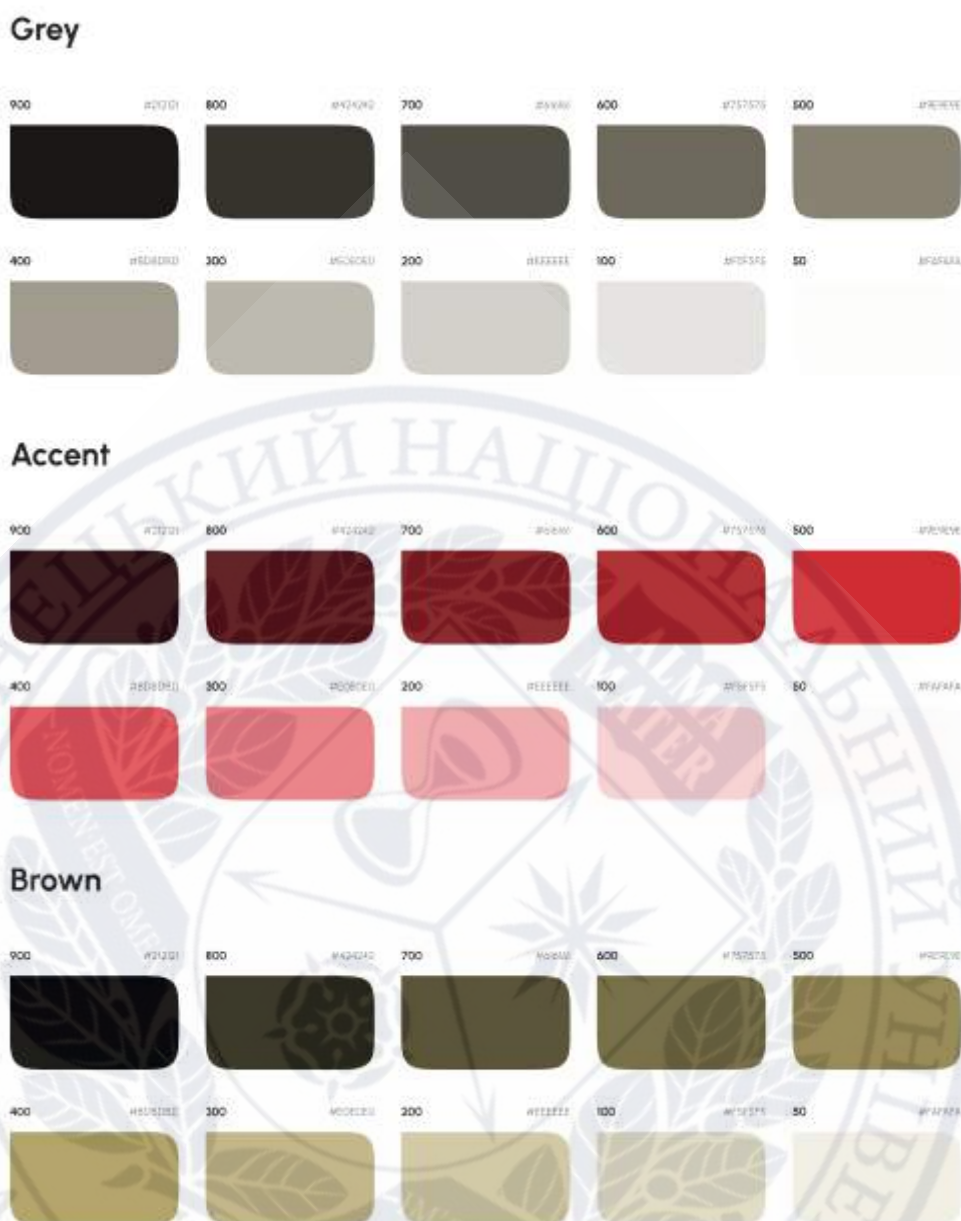


Рисунок 2.10 – Колірна палітра застосунку

Основним фоновим кольором застосунку обрано теплий бежевий відтінок (#F1EFE4), який використовується на всіх екранах. Вибір цього відтінку зумовлений прагненням зменшити зорове навантаження при тривалій взаємодії з контентом. Його асоціація з кольором книжкових сторінок сприяє створенню відповідної атмосфери читання [17]. Таке фонове рішення не відвертає увагу від основного контенту, водночас створюючи відчуття комфорту та візуальної гармонії.

Для акцентів використовуються більш насичені кольори, такі як темний коричневий (#191815) та ніжний кремовий (#E0DAC2). Темний колір

застосовується для тексту заголовків та основної навігаційної панелі, що дозволяє чітко виділяти важливу інформацію на фоні світлої палітри. Кремовий відтінок використовується для кнопок та інтерактивних елементів, що додає інтерфейсу м'яких, неагресивних візуальних акцентів [17]. Червоний колір (#E04D53) застосовується для сповіщень та кнопок підтвердження дій, що забезпечує їх високу помітність. Вибір цього відтінку обґрунтований його здатністю привертати увагу та асоціативним зв'язком з емоційною складовою обговорень. З метою уникнення візуального перевантаження інтерфейсу, червоний колір використовується дозовано, виключно для акцентування критично важливих елементів, що дозволяє зберегти загальну стилістичну цілісність [16].

Для текстового оформлення було обрано дві гарнітури: Bitter та Albra Grotesk. На рис. 2.11 продемонстровано застосування зазначених шрифтів. Гарнітура Bitter поєднує сучасний стиль із класичною елегантністю, що відповідає тематиці застосунку. Її чіткі лінії та висока читабельність роблять її оптимальною для відображення основного текстового контенту, не спричиняючи надмірного зорового навантаження.



Рисунок 2.11 – Типографіка Bitter та Albra Grotesk

Гарнітура Albra Grotesk використовується для підзаголовків, підписів до кнопок та невеликих інформаційних блоків. Її плавні лінії надають загальній стилістиці витонченості та візуальної завершеності. Для заголовків

використовується накреслення Bitter Bold, що забезпечує візуальне виділення ключових елементів. Основний текст набрано шрифтом Bitter Regular, характеристики якого (середня насиченість, оптимальний інтерліньяж) сприяють легкості читання. Для підписів та коментарів застосовується Albra Grotesk Medium, що створює помірний контраст та гармонійно інтегрується в загальну візуальну композицію.

Вибір графічних елементів також ґрунтується на принципах мінімалізму. Усі іконки характеризуються простою, лаконічною та легко впізнаваною формою, що сприяє швидкому розумінню їх функціонального призначення. На рис. 2.12 представлено ключові компоненти інтерфейсу, зокрема кнопки, поля введення, навігаційні панелі та картки книг.

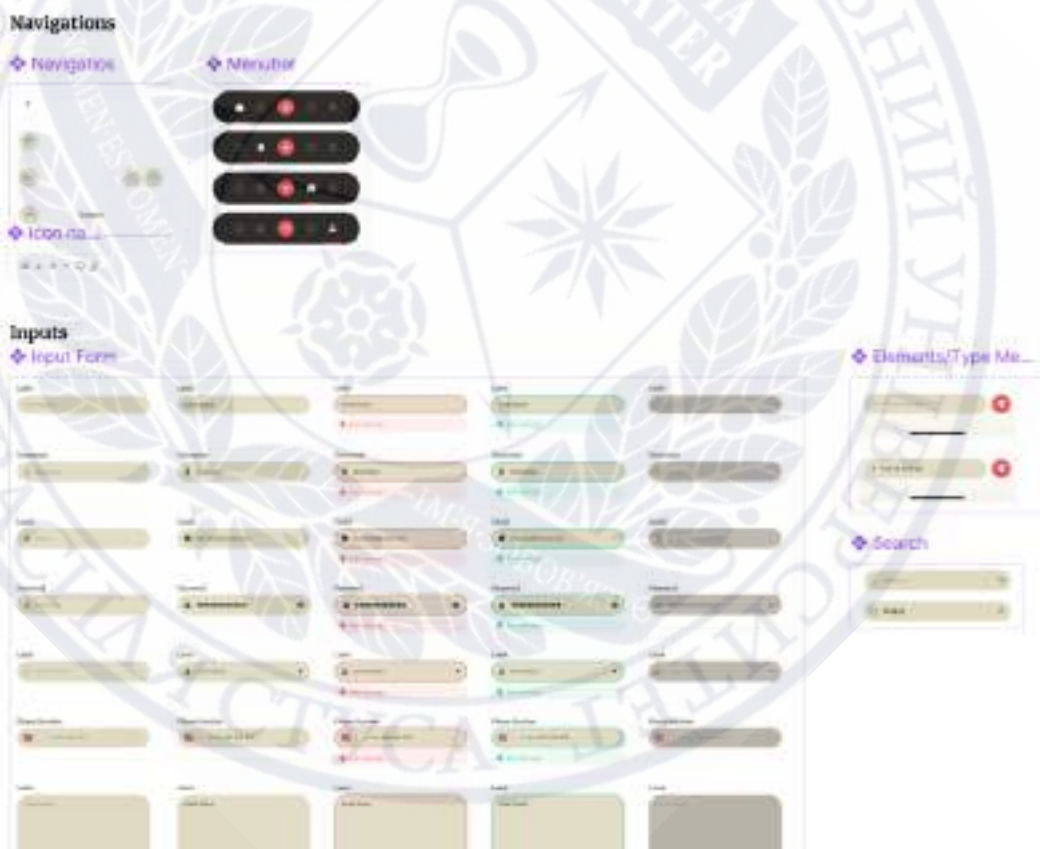


Рисунок 2.12 – Графічні елементи інтерфейсу

Кнопки реалізовані у вигляді блоків із заокругленими кутами та ледь помітною тінню, що надає їм візуальної легкості та забезпечує помітність на світлому фоні. Активні кнопки оформлені з використанням темно-коричневого фону та білого тексту, що створює високий контраст та чітко сигналізує про можливість взаємодії. Неактивні кнопки, навпаки, мають світлий фон та текст

зниженої контрастності, інформуючи користувача про тимчасову недоступність відповідних функцій. Особливу увагу приділено дизайну кнопок підтвердження та скасування дій: вони виконані у червоному та світло-кремовому кольорах відповідно, що візуально підкреслює їхню функціональну відмінність та сприяє швидкому розпізнаванню користувачем.

Для екрану сповіщень використано нейтральний світло-кремовий фон із червоним індикатором для позначення нових повідомлень. Таке рішення забезпечує комфортне сприйняття інформації та дозволяє користувачам швидко ідентифікувати нові події. Сповіщення представлені у вигляді вертикального списку з зазначенням імен користувачів та коротким описом відповідної дії [19].

Вибір колірної палітри, типографіки та графічних елементів здійснювався з урахуванням загальної стилістичної концепції застосунку та завдання створення комфортного середовища для взаємодії користувачів з контентом. Візуальний стиль базується на поєднанні теплих відтінків, плавних форм та легкої для сприйняття типографіки, що спрямовано на підвищення естетичної привабливості та ергономічності інтерфейсу.

2.3. Проектування адаптивного дизайну та взаємодії користувача

Проектування адаптивного дизайну для мобільного застосунку спільноти книголюбів становило значне завдання. Необхідно було забезпечити коректне відображення та функціонування інтерфейсу на широкому спектрі пристроїв, від компактних смартфонів до планшетів із великою діагоналлю екрана. Ключовим завданням було збереження інтуїтивності та зручності інтерфейсу незалежно від розміру екрана та його орієнтації. Для досягнення цієї мети було прийнято рішення використовувати гнучкі сітки (flexible grids) та відносні одиниці вимірювання. Цей підхід дозволяє елементам інтерфейсу динамічно змінювати свої розміри та взаємне розташування, адаптуючись до параметрів екрана та зберігаючи при цьому загальну структуру й візуальну цілісність. Застосування зазначених технік дозволило розробити дизайн, що забезпечує належне візуальне представлення та функціональність на різних пристроях [21].

Для забезпечення зручності та візуальної узгодженості інтерфейсу на екранах різних розмірів було застосовано технологію Flexbox. Flexbox надає ефективні інструменти для компоновання елементів, дозволяючи їм гнучко адаптуватися до доступного простору. Як приклад, у програмному коді головного екрану (BooksScreen) Flexbox використовується для структурування контейнера з картками книг:

```
booksContainer: {
  flexDirection: "row",
  flexWrap: "wrap",
  alignSelf: "flex-start",
  paddingHorizontal: width * 0.02,
},
book: {
  alignItems: "center",
  width: "50%", // Приблизно половина доступного місця в рядку
},
bookCover: {
  width: "90%",
  height: 90,
  borderRadius: width * 0.025,
  marginBottom: height * 0.005,
},
```

На головному екрані (BooksScreen) Flexbox забезпечує динамічне компоновання карток книг, які автоматично адаптуються до ширини та висоти дисплея, зберігаючи задане взаємне розташування. Це дозволяє підтримувати єдиний візуальний стиль та запобігати порушенню структури інтерфейсу при переході між пристроями з різними розмірами екранів, наприклад, зі смартфона на планшет [21]. На рис. 2.13 представлено приклад головного екрану із книжковими рекомендаціями, що демонструє адаптивність карток книг при зміні ширини екрану.

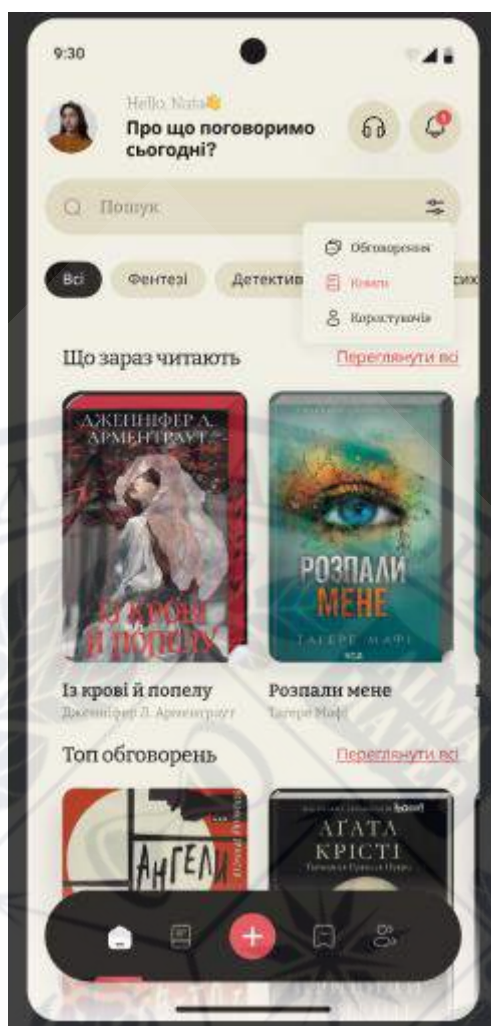


Рисунок 2.13 – Адаптивність головного екрану

При проектуванні головного екрану основну увагу було приділено динамічному розташуванню карток книг, забезпечуючи їх адаптацію до різних розмірів екранів. Для запобігання спотворенню обкладинок було використано функцію пропорційного масштабування зображень, що зберігає їх співвідношення сторін на пристроях із різними діагоналями екранів [22]. На рис. 2.14 продемонстровано реалізацію адаптивного компонування елементів на екрані обговорень. Візуальна структура цього екрану організована у вигляді вертикального блоку з динамічною зміною розмірів карток залежно від параметрів екрана.

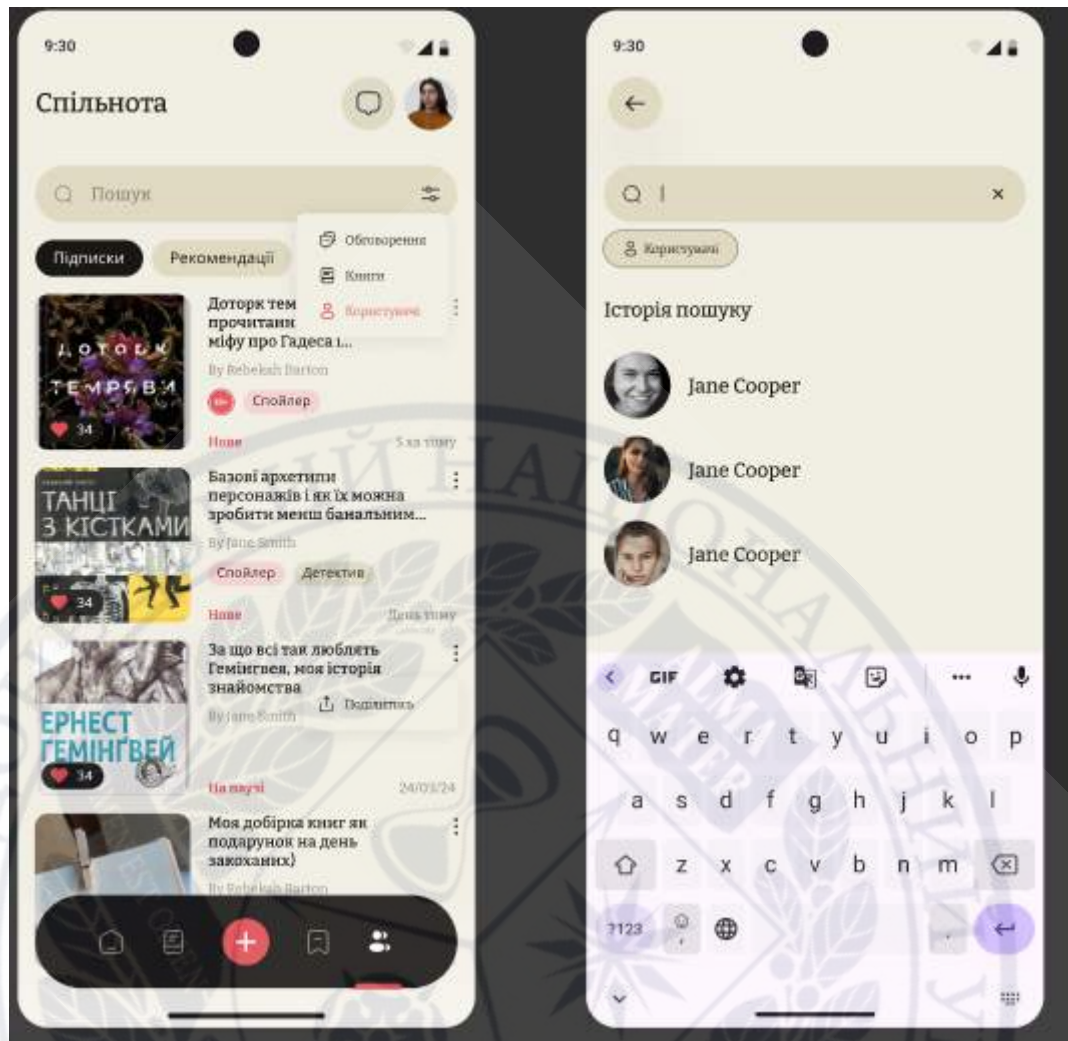


Рисунок 2.14 – Адаптивність екрану спільноти

Картки книг (або тем обговорень) представлені у вигляді сітки, що забезпечує достатній простір як для візуальних елементів (обкладинок), так і для текстового контенту. При зменшенні ширини екрана кількість карток у рядку автоматично скорочується для запобігання надмірному ущільненню елементів. Таке рішення дозволяє підтримувати баланс між візуальним комфортом та функціональністю інтерфейсу. Екран обговорень також був суттєво модифікований для забезпечення адаптивності. Система відображення коментарів реалізована з вертикальним розташуванням основних блоків у межах гнучкого контейнера. Це забезпечує автоматичну адаптацію компонування до доступного простору зі збереженням логічної послідовності контенту. Приклад стилізації елементів обговорення у файлі SavedScreen.js наведено нижче:

```
discussionItem: {
  backgroundColor: "#F1EFE4",
  borderRadius: 10,
```

```
padding: 15,
marginBottom: 15,
flexDirection: "row",
alignItems: "flex-start",
marginHorizontal: 16,
},
discussionImage: {
width: 120,
height: 120,
borderRadius: 8,
marginRight: 15,
},
```

Для екрану профілю користувача було розроблено два варіанти компонування, оптимізовані для портретної (вертикальної) та ландшафтної (горизонтальної) орієнтацій екрана. У портретній орієнтації елементи розташовуються послідовно: аватар, а потім основна інформація про користувача. У ландшафтній орієнтації блоки компонуються у два стовпці, що дозволяє більш ефективно використовувати доступний екранний простір [23]. Такий підхід сприяє збалансованому представленню текстової інформації та візуальних елементів, наприклад, обкладинок збережених книг. У файлі EditProfileScreen.js було реалізовано відповідні стилі для забезпечення гармонійного відображення профілю при будь-якій орієнтації пристрою:

```
profileImageContainer: {
alignItems: "center",
marginBottom: 130,
},
profileImage: {
width: 100,
height: 100,
borderRadius: 50,
},
inputContainer: {
marginBottom: 15,
bottom: 80,
},
```

При проектуванні екрану створення нової теми враховувалася необхідність забезпечення зручного введення тексту та завантаження зображень. Поля введення тексту реалізовані як гнучкі контейнери, здатні розширюватися по вертикалі відповідно до обсягу введенного тексту. Це дозволяє мінімізувати необхідність вертикальної прокрутки та забезпечує рівномірний розподіл простору між полями введення та інтерактивними кнопками. Для екрану збережених тем було застосовано сіткову структуру для відображення

зображень, яка автоматично адаптується до ширини екрана. Відображення обкладинок книг налаштоване таким чином, що їх висота залишається фіксованою, тоді як ширина може варіюватися залежно від кількості елементів у рядку. Реалізація в файлі `CollectionScreen.js` має такий вигляд:

```
booksContainer: {
  flexDirection: "row",
  flexWrap: "wrap",
  paddingHorizontal: 8,
},
bookItem: {
  width: "50%",
  paddingHorizontal: 8,
  marginBottom: 16,
},
bookCover: {
  width: "100%",
  height: 200,
  borderRadius: 8,
},
```

Навігаційна панель також була адаптована для коректного відображення на екранах різних розмірів. Кожна кнопка навігації має фіксовану висоту, тоді як її ширина може динамічно змінюватися для забезпечення зручності натискання, особливо на екранах невеликого розміру [21]. З цією метою застосовано техніку масштабування шрифтів та іконок, що дозволяє зберігати чіткість елементів незалежно від їх фактичного розміру. Приклад реалізації в файлі `SavedScreen.js` наведено нижче:

```
navigationBar: {
  flexDirection: "row",
  justifyContent: "space-around",
  alignItems: "center",
  backgroundColor: "#000",
  height: height * 0.1,
  width: width * 0.9,
  borderRadius: width * 0.1,
  position: "absolute",
  bottom: height * 0.02,
},
```

Використання адаптивних компонентів та технології Flexbox дозволило створити інтерфейс, що динамічно адаптується до розмірів екрана, забезпечуючи користувачам зручний доступ до всіх функціональних можливостей застосунку без порушення структурної та контентної цілісності. Застосування зазначених підходів також сприяє збереженню естетичної привабливості інтерфейсу на різних типах пристроїв.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНТЕРФЕЙСУ

3.1. Вибір інструментів і розробка інтерфейсу в коді

У процесі розробки мобільного застосунку "Book Talk" для спільноти книголюбів виникла необхідність вибору технологічної основи. Ключовою вимогою була кросплатформеність, що забезпечувала б належне функціонування застосунку на операційних системах iOS та Android без необхідності розробки окремих кодових баз. З огляду на це, було обрано фреймворк React Native, який дозволяє розробляти мобільні застосунки для обох платформ, використовуючи єдину кодову базу. React Native забезпечує не тільки візуальну відповідність нативним інтерфейсам, але й високу продуктивність та зручність використання, порівнянну з додатками, розробленими нативно для кожної платформи окремо. Таке рішення дозволило оптимізувати часові ресурси на розробку та забезпечити уніфікований користувацький досвід на різних пристроях [24].

Архітектура проєкту базується на компонентному підході, де кожен екран або функціональний блок реалізований як окремий компонент. Наприклад, головний екран представлений компонентом HomeScreen, екран з детальною інформацією про книгу – BookDetailsScreen, а екран сповіщень – NotificationScreen. Цей підхід забезпечує модульність: модифікація або додавання нової функціональності здійснюється шляхом роботи з ізольованими компонентами, що мінімізує вплив на інші частини системи. Це суттєво спрощує процес розробки та підтримки, а також сприяє підвищенню чистоти та зрозумілості коду [25].

Для стилізації компонентів було використано StyleSheet API, що є стандартним інструментом React Native. Даний API дозволяє визначати стилі засобами JavaScript, які компілюються в нативні еквіваленти під час виконання, забезпечуючи таким чином високу продуктивність візуалізації. Приклад визначення стилів у файлі HomeScreen.js наведено нижче:

```
const styles = StyleSheet.create({  
  safeArea: {
```

```

    flex: 1,
    backgroundColor: '#F1EFE4',  },
  container: {
    flex: 1,
    backgroundColor: '#F1EFE4',  },
  header: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    alignItems: 'center',
    marginBottom: height * 0.08,
    marginTop: height * 0.01,
  });
});

```

Навігація між екранами застосунку реалізована з використанням бібліотеки React Navigation, яка надає гнучку систему маршрутизації та підтримує різноманітні навігаційні патерни. У проєкті застосовано комбінацію стекової навігації (stack navigation) для послідовних переходів між екранами (наприклад, авторизація, реєстрація, деталі книги) та табової навігації (tab navigation) для доступу до основних розділів застосунку. Система навігації спроектована з метою забезпечення інтуїтивного переміщення користувача між різними функціональними блоками.

Для забезпечення чіткості графічних елементів на екранах з різною щільністю пікселів було прийнято рішення використовувати векторні зображення у форматі SVG. З цією метою було інтегровано бібліотеку react-native-svg, яка дозволяє рендерити іконки та інші графічні елементи без втрати якості при масштабуванні. Це є особливо актуальним для застосунків, орієнтованих на використання на широкому спектрі пристроїв з різними характеристиками дисплеїв. Використання SVG гарантує збереження чіткості та візуальної акуратності іконок незалежно від роздільної здатності екрана.

Реалізація адаптивного дизайну була одним із ключових аспектів розробки. За допомогою Dimensions API, що надається React Native, розміри елементів інтерфейсу обчислюються динамічно, базуючись на актуальних розмірах екрана конкретного пристрою [24]. Наприклад, у файлі BookScreen.js для адаптивного форматування навігаційної панелі використовується наступний підхід:

```

const { width, height } = Dimensions.get('window');
// ...
navigationBar: {

```

```

flexDirection: 'row',
justifyContent: 'space-around',
alignItems: 'center',
backgroundColor: '#000',
height: height * 0.1,
width: width * 0.9,
borderRadius: width * 0.1,}

```

При розробці форм авторизації та реєстрації особливу увагу було приділено деталям інтерфейсу, оскільки ці екрани є точками частої взаємодії з користувачем. Замість стандартних полів введення було створено кастомні компоненти, які забезпечують покращений візуальний стиль та обробку різних станів (наприклад, коректне/помилкове введення, активність поля). Для полів введення, що містять іконки, компоненти `TextInput` та `SVG-іконки` було об'єднано в єдиний контейнер. Це дозволило забезпечити єдиний візуальний стиль для всіх форм у застосунку. В результаті, текстові поля мають цілісний вигляд, що спрощує користувачам процес введення даних та орієнтацію в інтерфейсі [24]. Для реалізації плавних анімацій та переходів між станами інтерфейсу було використано `Animated API`, що є частиною ядра `React Native`. Цей інструмент дозволяє створювати анімації, що виконуються в нативному потоці (`native thread`), не блокуючи основний потік `JavaScript`, що забезпечує високу продуктивність навіть на пристроях з обмеженими ресурсами. Анімації було застосовано для покращення візуального відгуку інтерфейсу на дії користувача [26, 41].

Модальні вікна, які використовуються для відображення додаткової інформації або опцій дій, реалізовані за допомогою стандартного компонента `Modal` з `React Native` [25]. Наприклад, у файлі `SavedScreen.js` для створення діалогового вікна з опціями дій щодо елемента обговорення використовується наступна реалізація:

```

<Modal
  animationType="slide"
  transparent={true}
  visible={isThreeDotsMenuVisible}
  onRequestClose={closeThreeDotsMenu}
>
  <View style={styles.centeredView}>
    <View style={styles.modalView}>
      <TouchableOpacity

```

```

        style={styles.modalOption}
        onPress={() => {
            console.log(`Поділитись обговоренням з ID:
${selectedDiscussionItemId}`);
            closeThreeDotsMenu();
        }}
    >
    <Share width={20} height={20} style={styles.modalIcon}
/>
    <Text style={styles.modalText}>Поділитись</Text>
</TouchableOpacity>
    { /* Інші опції... */ }
</View>
</View>
</Modal>

```

Типографіці та оформленню тексту було приділено значну увагу, враховуючи специфіку цільової аудиторії – книголюбів. Для основного тексту обрано гарнітуру Bitter, що характеризується високою читабельністю на екранах різних розмірів та асоціюється з традиційними книжковими шрифтами. Для заголовків та акцентних елементів використано гарнітуру Albra, яка надає інтерфейсу сучасного вигляду, зберігаючи при цьому зв'язок з книжковою естетикою [28].

Початково для реалізації списків книг та обговорень використовувався компонент ScrollView. Однак, зі збільшенням обсягу даних такий підхід призводив до зниження продуктивності застосунку. Тому було прийнято рішення перейти на використання компонента FlatList. FlatList оптимізує відображення списків шляхом рендерингу лише тих елементів, які є видимими на екрані в поточний момент, та поступового довантаження інших елементів під час прокручування. Це дозволило значно знизити навантаження на оперативну пам'ять та забезпечити більш плавну прокрутку списків.

Стосовно навігаційної панелі, стандартна реалізація табової панелі з React Navigation не повною мірою відповідала дизайнерській концепції, тому було розроблено кастомне рішення. Панель було розміщено в нижній частині екрана та зафіксовано за допомогою абсолютного позиціонування. Для забезпечення чіткої візуальної індикації поточного розділу було додано стильові маркери для активних вкладок. В результаті, навігаційна панель гармонійно інтегрувалася в

загальний дизайн, набувши більш сучасного вигляду та покращивши зручність використання.

Для реалізації функціоналу фільтрації книг за категоріями було розроблено систему, що використовує хук `useState` для управління станом компонента та функції фільтрації, які обробляють масив даних відповідно до вибору користувача. Наприклад, у файлі `Home_in.js` реалізовано логіку фільтрації книг за жанрами наступним чином:

```
const filterBooks = (genre) => {
  const filtered =
    genre === "Bci"
      ? allBooks.slice(0, 4)
      : allBooks.filter((book) => book.genre ===
genre).slice(0, 4);
  setDisplayedBooks(filtered);};
```

Організація кодової бази проєкту спрямована на забезпечення максимальної простоти та зрозумілості. Компоненти розроблено за принципом повторного використання (*reusability*), що дозволяє їх легко інтегрувати в різні частини застосунку без необхідності дублювання коду. Функціональність застосунку декомпозована на логічні модулі, що дозволяє вносити зміни або додавати нові можливості ізольовано, мінімізуючи вплив на інші частини системи. Такий підхід забезпечує високий рівень масштабованості та гнучкості, дозволяючи легко розширювати функціональні можливості застосунку шляхом додавання нових компонентів або екранів без ризику порушення наявної функціональності.

3.2. Проведення юзабіліті-тестування та аналіз зворотного зв'язку

Після розробки першої бета-версії мобільного застосунку "Book Talk" було проведено юзабіліті-тестування з метою оцінки зручності його використання. У тестуванні взяли участь 15 респондентів віком від 18 до 55 років. Вибірка включала як технічно досвідчених користувачів, так і осіб з початковим рівнем володіння мобільними додатками. Такий склад учасників дозволив отримати різнобічний зворотний зв'язок та оцінити застосунок з позицій різних груп користувачів. Тестування проводилося за заздалегідь розробленими сценаріями,

що включали виконання таких завдань: реєстрація нового облікового запису, пошук книги, додавання книги до колекції, створення обговорення, залишення коментаря та зміна налаштувань профілю. У процесі тестування фіксувалися не тільки кількісні показники (час виконання завдань, кількість кліків), але й якісні дані: емоційні реакції, вербальні коментарі та загальні враження респондентів від взаємодії із застосунком. Такий підхід дозволив зібрати не лише кількісні метрики, а й якісні дані, що сприяло виявленню сильних сторін інтерфейсу та зон, які потребують удосконалення.

Під час першого етапу тестування було виявлено низку проблемних аспектів. Зокрема, користувачі вказували на труднощі з навігацією: семеро з п'ятнадцяти респондентів зазначили незрозумілість механізму повернення на головний екран з екрана деталізації книги. Це вказувало на необхідність внесення змін до навігаційних елементів. За результатами аналізу зворотного зв'язку було прийнято рішення інтегрувати більш помітну кнопку "Назад" у вигляді стрілки на верхній панелі. Додатково було впроваджено візуальні індикатори для покращення орієнтації користувача в структурі застосунку. Ці зміни були спрямовані на підвищення зрозумілості та зручності навігації [30]. Значна частина зауважень стосувалася функціональності пошуку книг: користувачі висловили потребу в розширених можливостях фільтрації та сортування результатів. У відповідь на ці зауваження, функціонал пошуку було доопрацьовано шляхом реалізації додаткових опцій фільтрації за категоріями (у файлі `search.js`), що позитивно вплинуло на користувацький досвід.

Респонденти також вказали на проблеми з розміром елементів інтерфейсу на екрані профілю, особливо при використанні пристроїв з невеликими екранами. Було встановлено, що інтерактивні елементи, зокрема кнопки, мали недостатні розміри для комфортного натискання. Для вирішення цієї проблеми було збільшено розміри кнопок та впроваджено адаптивне масштабування елементів інтерфейсу. Це дозволило підвищити зручність та зрозумілість інтерфейсу для користувачів різних пристроїв. Тестування також виявило проблеми з читабельністю тексту на певних екранах застосунку, особливо серед

користувачів старшої вікової групи. Зокрема, було відзначено недостатній контраст тексту опису книги відносно фону на екрані деталей. У відповідь було переглянуто колірну схему для забезпечення вищого рівня контрастності та збільшено розмір шрифту для описових блоків тексту. Ці зміни позитивно вплинули на загальну доступність застосунку для різних категорій користувачів [32].

Труднощі у взаємодії користувачів виникали і з модальними вікнами: частина респондентів не одразу розуміла механізм їх закриття, що викликало негативні емоції, особливо з вікнами для додавання книги до колекції. Для вирішення цієї проблеми було збільшено візуальну помітність кнопки закриття, а також оновлено дизайн модальних вікон для покращення їх зрозумілості та логічності структури. Це дозволило спростити взаємодію з модальними вікнами.

Аналіз процесу реєстрації та входу виявив необхідність у наданні додаткових підказок користувачам. У відповідь було інтегровано пояснювальні тексти та механізми валідації форм для запобігання помилкам при введенні даних. Також було реалізовано функцію відображення/приховування пароля, яку респонденти позитивно оцінили як суттєве покращення зручності:

```
<View style={styles.inputContainer}>
  <Text style={styles.label}>Пароль</Text>
  <View style={styles.inputWithIcon}>
    <Password style={styles.icon} resizeMode="contain" />
    <TextInput
      style={[styles.inputWithLeftIcon, styles.fontBitter]}
      placeholder="Введіть Ваш пароль"
      placeholderTextColor="#888272"
      secureTextEntry={!isPasswordVisible}
      value={password}
      onChangeText={setPassword}
    />
    <TouchableOpacity
      style={styles.passwordVisibilityButton}
      onPress={togglePasswordVisibility}
    >
      <Text style={styles.passwordVisibilityText}>
        {isPasswordVisible ? 'Hide' : 'Show'}
      </Text>
    </TouchableOpacity>
  </View>
</View>
```

Ключові проблеми, виявлені під час першого етапу юзабіліті-тестування, кількісно представлені на рис. 3.1. Як видно з діаграми, найбільша кількість користувачів (15) вказала на неоднозначність у позиціонуванні соціальних функцій відносно функцій каталогізації, що потребувало подальшого аналізу переваг різних груп користувачів. Також значними проблемами виявились навігація (зокрема, повернення на головний екран – 7 користувачів) та фільтрація книг (6 користувачів).



Рисунок 3.1 – Ключові проблеми, виявлені під час юзабіліті-тестування

Після внесення змін за результатами першого етапу було проведено повторне юзабіліті-тестування з тією ж групою респондентів для оцінки ефективності впроваджених удосконалень. Результати повторного тестування продемонстрували значне покращення ключових метрик користувацького досвіду: середній час виконання завдань скоротився на 28%, а кількість помилок, допущених користувачами, зменшилася на 37%. Загальний рівень задоволеності застосунком, оцінений за 10-бальною шкалою, зріс з 6.8 до 8.4 балів. Динаміку змін ключових метрик ефективності використання представлено на рис. 3.2.



Рисунок 3.2 – Вплив внесених змін на ключові метрики ефективності використання застосунку

Додатково до формального тестування було реалізовано функціонал для збору зворотного зв'язку безпосередньо в застосунку. Користувачі отримали можливість повідомляти про проблеми або надсилати пропозиції через спеціальний екран зворотного зв'язку (компонент Feedback.js). Це забезпечило прямий канал комунікації між розробниками та користувачами. Такий підхід сприяє оперативному виявленню проблемних аспектів та своєчасному внесенню необхідних коректив [33].

Аналіз отриманого зворотного зв'язку дозволив ідентифікувати низку потенційних напрямків для подальшого розвитку функціоналу. Зокрема, користувачі позитивно оцінили функцію збереження обговорень, але висловили побажання щодо розширення можливостей їх організації, наприклад, шляхом створення користувацьких категорій. Цю пропозицію було враховано та включено до плану подальших оновлень. Іншою популярною пропозицією стала інтеграція функції поширення обговорень через соціальні мережі. З огляду на доцільність, дану функціональність було оперативно реалізовано, надавши користувачам можливість легко ділитися цікавими обговореннями.

Важливим спостереженням стало виявлення вікових відмінностей у перевагах щодо використання функцій: користувачі віком 18–30 років частіше зверталися до функцій соціальної взаємодії (коментування, оцінки), тоді як респонденти старшої вікової групи (31–55 років) виявляли більший інтерес до функцій каталогізації та пошуку книг. Відмінності у вподобаннях користувачів різних вікових груп щодо функціоналу застосунку ілюструє рис. 3.3. Ці дані дозволили глибше зрозуміти потреби різних сегментів цільової аудиторії та скоригувати пріоритети подальшого розвитку застосунку. Також було вдосконалено навігацію за допомогою клавіатури та додано текстові альтернативи для всіх візуальних елементів з метою підвищення доступності [30].

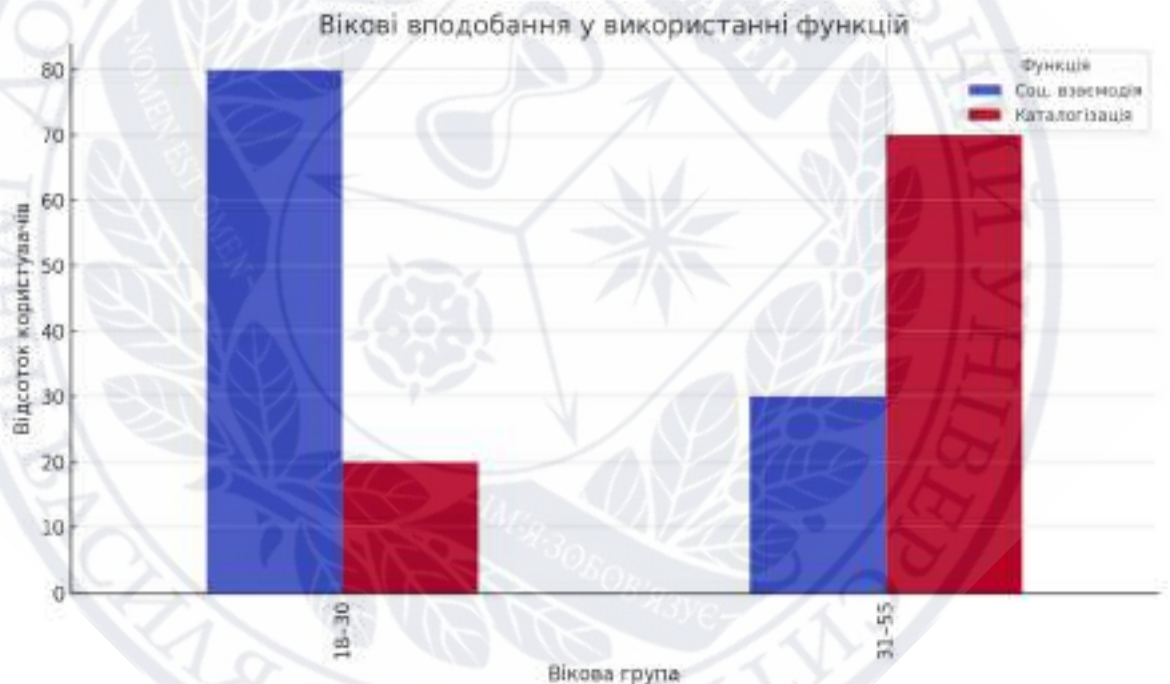


Рисунок 3.3 – Вікові вподобання користувачів у використанні ключових функцій застосунку

Аналіз метрик використання застосунку продемонстрував, що середня тривалість сесії становить близько 15 хвилин, а найбільш відвідуваними розділами є обговорення книг та пошук нової літератури. Ці дані були використані для оптимізації інтерфейсу шляхом розміщення найбільш

затребуваних функцій у легкодоступних місцях та спрощення навігаційних шляхів до них [31].

Значна кількість користувачів висловила побажання щодо впровадження темної теми інтерфейсу, особливо для комфортного читання в умовах низької освітленості. Початково дана функція не була запланована для першого релізу, однак через численні запити її було включено до плану подальшого розвитку. Для попередньої оцінки зручності темної теми було створено декілька прототипів, які були продемонстровані тестовій групі. Зворотний зв'язок був переважно позитивним: респонденти відзначили естетичну привабливість та покращену зручність читання в темному режимі. Таким чином, реалізація темної теми була визначена як один із пріоритетів для наступного оновлення застосунку.

Проведений комплекс робіт підтвердив, що регулярне проведення юзабіліті-тестування та систематичний аналіз зворотного зв'язку є невід'ємними інструментами для створення продукту, який максимально відповідає потребам користувачів та забезпечує високий рівень зручності використання.

3.3. Оптимізація дизайну та продуктивності інтерфейсу

У процесі розробки мобільного застосунку "Book Talk" особливу увагу було приділено оптимізації дизайну та продуктивності інтерфейсу. Було проаналізовано зворотний зв'язок, отриманий під час юзабіліті-тестування, та враховано технічні характеристики широкого спектра цільових пристроїв, від моделей попередніх поколінь до сучасних флагманів. Основне завдання полягало в забезпеченні високої швидкодії та естетичної привабливості застосунку на всіх підтримуваних пристроях. Інтерфейс повинен був характеризуватися не тільки візуальною якістю, але й інтуїтивною зрозумілістю, що дозволяло б користувачам легко виконувати ключові дії, такі як пошук книги, додавання її до колекції або залишення коментаря.

На початкових етапах розробки для відображення списків книг та обговорень використовувався компонент ScrollView. Однак, зі збільшенням

обсягу даних було виявлено зниження продуктивності, особливо на пристроях з обмеженими обчислювальними ресурсами. Проблема полягала в тому, що ScrollView здійснював рендеринг усіх елементів списку одночасно, включаючи ті, що знаходилися поза видимою областю екрана. Для вирішення цієї проблеми було здійснено перехід на компонент FlatList. FlatList реалізує механізм віртуалізації списків, рендерячи лише ті елементи, які знаходяться в поточній видимій зоні, та динамічно довантажуючи інші елементи під час прокручування. Це дозволило суттєво зменшити навантаження на оперативну пам'ять та забезпечити вищу плавність і швидкість прокручування [34].

З метою зменшення розміру застосунку та підвищення швидкодії було проведено оптимізацію графічних ресурсів. Усі растрові зображення, де це було доцільно, було замінено на векторні файли у форматі SVG. Це стосувалося, зокрема, іконок навігаційної панелі, кнопок та інших елементів інтерфейсу. Перевага SVG полягає в їх меншому розмірі порівняно з растровими аналогами, здатності до масштабування без втрати якості та забезпеченні чіткого відображення на екранах з різною щільністю пікселів. Завдяки цьому вдалося не тільки зменшити загальний розмір інсталяційного пакета застосунку, але й покращити візуальну якість та акуратність інтерфейсу [35].

Для прискорення завантаження контенту та зменшення споживання мережевого трафіку було оптимізовано процес завантаження обкладинок книг. Початково всі зображення завантажувалися одночасно при відкритті екрана, що призводило до затримок в роботі інтерфейсу, особливо при великій кількості елементів. Було впроваджено механізм відкладеного завантаження (lazy loading), за яким обкладинка книги завантажується лише тоді, коли відповідний елемент списку потрапляє у видиму область екрана. Це дозволило зменшити споживання трафіку та прискорити початкове завантаження екранів. Для прискорення повторного доступу до зображень було реалізовано механізм кешування, що дозволяє завантажувати їх з локального сховища замість повторного запиту до сервера.

Наступним напрямком оптимізації стало вдосконалення системи навігації. За результатами юзабіліті-тестування було виявлено, що частина користувачів відчувала труднощі з орієнтацією в структурі застосунку та визначенням шляху повернення до попередніх екранів. Для вирішення цієї проблеми було розроблено уніфіковану навігаційну панель, яка чітко індикує поточний активний розділ та забезпечує швидкий доступ до ключових функцій застосунку. Для індикації активного розділу використовувався стан компонента (`activeTab`), що оновлювався при переході між екранами.

Початково анімаційні ефекти в застосунку виконувалися в основному потоці JavaScript, що призводило до зниження продуктивності інтерфейсу, особливо під час одночасного завантаження значних обсягів даних. Для забезпечення плавності анімацій та уникнення затримок було прийнято рішення перенести їх виконання на нативний потік за допомогою опції `useNativeDriver: true`. Це дозволило виконувати анімації незалежно від основного потоку JavaScript, забезпечивши їх плавність та високу швидкість відгуку. Для покращення сприйняття процесу завантаження даних та зменшення суб'єктивного часу очікування було впроваджено патерн скелетного завантаження (`skeleton loading`). Цей патерн передбачає відображення тимчасових плейсхолдерів (контурів елементів інтерфейсу, таких як картки книг, коментарі тощо) на місці контенту, що завантажується. Це створює у користувача враження негайного завантаження та готовності інтерфейсу, навіть якщо фактичне завантаження даних ще триває. Таким чином, зменшується відчуття "зависання" застосунку та підвищується загальне сприйняття швидкодії. Це рішення не тільки скорочує суб'єктивний час очікування, але й забезпечує візуальну цілісність інтерфейсу, дозволяючи користувачеві заздалегідь розуміти структуру майбутнього контенту. В результаті, взаємодія із застосунком сприймається як більш швидка та плавна, навіть якщо об'єктивний час завантаження залишається незмінним [37].

Оптимізація дизайну також охоплювала перегляд колірної схеми та типографіки з метою підвищення читабельності тексту. На основі зворотного

зв'язку від користувачів було встановлено, що тривале використання застосунку спричиняло зорову втому у деяких респондентів. Для усунення цієї проблеми було скориговано контрастність між текстом та фоном, збільшено розмір базового шрифту з 14 до 16 пунктів та оптимізовано міжрядковий інтервал, що суттєво покращило сприйняття текстової інформації.

Розробка модульної системи компонентів стала наступним етапом оптимізації, спрямованим як на покращення процесу розробки, так і на підвищення продуктивності застосунку. Було створено бібліотеку уніфікованих компонентів багаторазового використання (наприклад, кнопки, текстові поля, картки книг), що забезпечило візуальну консистентність дизайну та дозволило скоротити загальний обсяг кодової бази. Такий підхід також спростив процес внесення майбутніх змін, оскільки модифікація окремого компонента не впливає на інші частини системи. Значне підвищення продуктивності було досягнуто шляхом оптимізації життєвого циклу компонентів та запобігання надлишковим операціям перерендерингу. З цією метою було застосовано техніки мемоізації, зокрема `React.memo`, хуки `useMemo` та `useCallback`, які дозволяють уникнути повторних обчислень та оновлень інтерфейсу, якщо відповідні вхідні дані (`props` або залежності) не змінилися. Ефект від застосування цих технік є особливо помітним на екранах з великою кількістю динамічних елементів, наприклад, на екрані обговорень книги [38].

З метою покращення користувацького досвіду (UX) було впроваджено концепцію мікровзаємодій – коротких анімаційних ефектів, що надають користувачеві візуальний зворотний зв'язок на його дії. Наприклад, при натисканні на кнопку "лайк" відбувається її незначне масштабування та зміна кольору, що створює відчуття тактильної взаємодії з елементом інтерфейсу. Мікровзаємодії було реалізовано з урахуванням вимог до продуктивності, щоб мінімізувати навантаження на пристрій та забезпечити плавність анімацій навіть на менш потужних смартфонах.

Під час юзабіліті-тестування було виявлено, що форми реєстрації та зворотного зв'язку спричиняли певні труднощі у користувачів. Респонденти

допускали помилки при заповненні, а в деяких випадках не повністю розуміли, яку інформацію необхідно вводити. Для спрощення процесу заповнення форм та підвищення його зрозумілості було впроваджено механізм миттєвої валідації полів. Застосунок тепер негайно індикує помилки введення та надає підказки щодо їх виправлення. Наприклад, при введенні електронної адреси без символу «@» система негайно сигналізує про помилку. Додатково було оптимізовано взаємодію з віртуальною клавіатурою: реалізовано автоматичне прокручування екрана до активного поля введення та автоматичний перехід фокуса до наступного поля після заповнення поточного. Ці заходи зробили процес заповнення форм більш плавним та інтуїтивно зрозумілим, зменшуючи ймовірність виникнення помилок та негативних емоцій у користувачів [38].

З метою підвищення доступності застосунку для користувачів з різними потребами було реалізовано підтримку динамічного масштабування шрифтів. Це дозволяє інтерфейсу застосунку адаптуватися до системних налаштувань розміру шрифту, забезпечуючи комфортне читання для осіб з вадами зору. Також було додано семантичні атрибути до всіх інтерактивних елементів, що забезпечує їх коректну інтерпретацію технологіями читання з екрана (screen readers).

Оптимізаційні заходи також охопили архітектуру управління станом (State Management) застосунку. Початкова реалізація, заснована на використанні локального стану компонентів, призводила до труднощів із синхронізацією даних між різними екранами. Для вирішення цієї проблеми було впроваджено централізоване управління станом за допомогою Context API. Це дозволило зменшити кількість надлишкових запитів до сервера та забезпечити консистентність даних у межах усього застосунку [40].

З метою покращення загального користувацького досвіду було оптимізовано анімації переходів між екранами. Замість різких змін стану інтерфейсу було реалізовано плавні анімаційні переходи, що відповідають контексту взаємодії. Наприклад, при відкритті екрана з детальною інформацією про книгу використовується плавна анімація появи знизу, що створює ефект

"занурення" в контент. Ці анімації було реалізовано з урахуванням вимог до продуктивності для уникнення додаткового навантаження на пристрій [39].

Ефективність проведених оптимізаційних заходів була підтверджена кількісними метриками продуктивності: час до інтерактивності (Time to Interactive, TTI) для ключових екранів застосунку скоротився на 35%; споживання оперативної пам'яті зменшилося на 28%; розмір інсталяційного пакета було зменшено на 22%. Ці вдосконалення дозволили забезпечити стабільну та плавну роботу застосунку навіть на мобільних пристроях середнього цінового сегмента, що відповідало вимогам проекту щодо охоплення максимально широкої аудиторії.

Загалом, комплексний процес оптимізації дизайну та продуктивності інтерфейсу дозволив створити мобільний застосунок, який не тільки відповідає сучасним стандартам UX/UI, але й характеризується високою швидкістю та плавністю роботи на широкому спектрі мобільних пристроїв. Інтеграція принципів оптимізації на ранніх етапах життєвого циклу розробки дозволила уникнути необхідності масштабного рефакторингу на пізніх стадіях та заклала надійну основу для подальшого розвитку та масштабування застосунку.

ВИСНОВКИ

У рамках дипломної роботи було розроблено мобільний застосунок "Book Talk", призначений для спільноти книголюбів та обміну читацькими враженнями. Застосунок розроблено з урахуванням сучасних стандартів дизайну та орієнтовано на задоволення потреб цільової аудиторії. Проведений попередній аналіз підтвердив актуальність розробки такої платформи: було встановлено, що 78% опитаних книголюбів виявляють зацікавленість у спеціалізованому просторі для обговорення прочитаних творів, а 82% респондентів висловили бажання вести облік прочитаних книг. Це підтвердило потенціал "Book Talk" як зручної платформи для читання, обговорення книг та отримання рекомендацій.

З метою визначення конкурентних переваг "Book Talk" було проведено аналіз існуючих популярних книжкових застосунків, таких як Goodreads, Bookly та Wattpad. Аналізувалися як сильні сторони цих платформ, так і аспекти, що викликають труднощі у користувачів. Результати аналізу стали основою для формування функціональних вимог до "Book Talk", при цьому основний акцент було зроблено на задоволенні потреб української аудиторії книголюбів. Реалізований функціонал "Book Talk" дозволяє користувачам створювати персональні колекції книг, залишати оцінки та рецензії, а також брати участь в обговореннях прочитаних творів. Для забезпечення комфортного обговорення було впроваджено функцію маркування спойлерів, що дозволило зробити дискусії більш безпечними для користувачів, які ще не завершили читання відповідної книги.

При розробці "Book Talk" значну увагу було приділено створенню зручного та візуально привабливого інтерфейсу. Дизайн інтерфейсу проектувався з розрахунком на інтуїтивне розуміння користувачем навігаційних шляхів та способів взаємодії. Ключові розділи застосунку, такі як головний екран, екран обговорень, профіль користувача та розділ збережених обговорень, розроблялися з акцентом на забезпечення інтуїтивної навігації. Колірна палітра була ретельно підібрана: теплий бежевий фон (#F1EFE4) створює асоціацію з

книжковими сторінками та сприяє комфортному читанню, тоді як темні відтінки використовуються для акцентування та покращення читабельності тексту. Метою такого дизайнерського рішення було створення атмосфери, що нагадує взаємодію з фізичною книгою.

Розробка інтерфейсу здійснювалася з використанням фреймворку React Native, що забезпечило створення кросплатформенного рішення, сумісного з операційними системами iOS та Android. Застосування модульного підходу, StyleSheet API, бібліотеки React Navigation та інструментів для роботи з SVG-графікою сприяло досягненню оптимальної продуктивності та високої естетичної якості застосунку. Адаптивний дизайн, реалізований за допомогою Dimensions API, забезпечує коректне відображення інтерфейсу на пристроях з різними розмірами екранів.

Проведене юзабіліті-тестування за участю 15 респондентів різних вікових категорій дозволило виявити низку проблемних аспектів, які були успішно усунені на етапі доопрацювання. Зокрема, було вдосконалено навігаційну систему, розширено функціонал пошуку та фільтрації книг, оптимізовано читабельність текстового контенту та покращено дизайн модальних вікон. Повторне тестування, проведене після внесення відповідних змін, зафіксувало значне покращення показників користувацького досвіду: середній час виконання завдань скоротився на 28%, а загальний рівень задоволеності застосунком зріс з 6.8 до 8.4 балів за 10-бальною шкалою.

Заходи з оптимізації дизайну та продуктивності інтерфейсу включали впровадження техніки віртуалізації списків за допомогою компонента FlatList, заміну растрових зображень векторними SVG-файлами, реалізацію відкладеного завантаження контенту та перенесення анімацій на нативний потік. Ці заходи призвели до значного покращення загальної продуктивності застосунку: час до інтерактивності (TTI) скоротився на 35%, споживання оперативної пам'яті зменшилося на 28%, а розмір інсталяційного пакета – на 22%.

Завдяки застосуванню системного підходу до процесу розробки, глибокому аналізу потреб користувачів та інтеграції сучасних технологій було

створено мобільний застосунок, що відповідає вимогам цільової аудиторії, характеризується інтуїтивно зрозумілим інтерфейсом та забезпечує високу продуктивність на широкому спектрі мобільних пристроїв. Розроблений застосунок "Book Talk" формує комфортне середовище для спілкування книголюбів, сприяє розвитку літературної спільноти та стимулює читацьку активність, що повністю відповідає основній меті, поставленій у дипломній роботі.



СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Book Girls' Guide. 2024 Reader Survey Results. URL: <https://bookgirlsguide.com/2024-reader-survey-results/> (дата звернення: 11.05.2025).
2. Statista. Digital Publishing Product Format Users in the USA. URL: <https://www.statista.com/statistics/455743/digital-publishing-product-format-users-digital-market-outlook-usa/> (дата звернення: 11.05.2025).
3. Penguin Random House. From the Reader Lounge: A Dive into Avid Readers and Book Clubs. URL: <https://authornews.penguinrandomhouse.com/from-the-reader-lounge-a-dive-into-avid-readers-and-book-clubs/> (дата звернення: 11.05.2025).
4. Bates C. C., Klein A., Schubert B. E-Books and E-Book Apps: Considerations for Beginning Readers // The Reading Teacher. – 2016. – Vol. 70, no. 4. – P. 401–411. – DOI: <https://doi.org/10.1002/trtr.1543> (дата звернення: 11.05.2025).
5. Liu H., Zhang M., Chen X. Exploring the Influential Factors on Readers' Continuance Intentions of E-Book Apps: Personalization, Usefulness, Playfulness, and Satisfaction // Frontiers in Psychology. – 2021. – Vol. 12. – DOI: <https://doi.org/10.3389/fpsyg.2021.640110> (дата звернення: 11.05.2025).
6. Tram M. Firebase / Mai Tram; Centria University of Applied Sciences. 2019. URL: <https://urn.fi/URN:NBN:fi:amk-2019112823179> (дата звернення: 11.05.2025).
7. Megantoro P., Widyawan, Sutanto D., Dzikrullah M. Instrumentation System for Data Acquisition and Monitoring of Hydroponic Farming Using ESP32 via Google Firebase // Indonesian Journal of Electrical Engineering and Computer Science. – 2022. – Vol. 27, no. 1. – P. 52–61. – DOI: <https://doi.org/10.11591/ijeecs.v27.i1.pp52-61> (дата звернення: 11.05.2025).
8. Багвані М. К., Тіварі В. К., Шарма Р. Deploying a Web Application on AWS Amplify: A Comprehensive Guide // International Journal of Progressive Research in Engineering Management and Science. – 2024. – Т. 4, № 8. – С. 538–544. – DOI: <https://doi.org/10.58257/IJPREMS35750> (дата звернення: 11.05.2025).
9. Беседін Д. А. Дослідження функціональних можливостей UI/UX інтерфейсів для мобільних додатків: магістерська атестаційна робота. Харків:

- Харківський національний університет радіоелектроніки, 2020. 57 с. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/d82308d5-d28f-4eef-844c-0cbf2f3fc432/content> (дата звернення: 11.05.2025).
10. Антипов А. К. Мобільний додаток для читання електронних книжкових видань з деталізацією процесу тестування користувацького інтерфейсу: дипломний проєкт на здобуття ступеня бакалавра. – Київ: КПІ ім. Ігоря Сікорського, Навчально-науковий видавничо-поліграфічний інститут, 2023. 68 с. URL: <https://ela.kpi.ua/server/api/core/bitstreams/7fc183c8-cc14-4e39-957a-f6e94e50afaf/content> (дата звернення: 11.05.2025).
 11. Горошко І., Горошко В., Горошко А. Мобільні додатки в освіті: розробка інтелектуального Android-застосунку для поліпшення інформаційної доступності розкладу занять // Measuring and Computing Devices in Technological Processes. – 2024. – № 1. – С. 13–20. – DOI: <https://doi.org/10.31891/2219-9365-2024-77-2> (дата звернення: 11.05.2025).
 12. Бикович Т. В. Дизайн користувацьких інтерфейсів мобільних застосунків. Розробка дизайну застосунку для покращення ментального здоров'я : кваліфікаційна робота магістра : спеціальність 022 «Дизайн» / Національна академія керівних кадрів культури і мистецтв. Київ, 2023. 55 с. URL: <https://elib.nakkkim.edu.ua/bitstream/handle/123456789/5314/bikovich.pdf?sequence=1&isAllowed=y> (дата звернення: 11.05.2025).
 13. Чеботарьова І. Б., Черкашина Г. І. Основні тренди UI/UX дизайну 2024 року: матеріали молодіжної школи-семінару. Харків: ХНУРЕ, 2024. С. 40–47. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/1dbf3811-3de3-4599-90aa-1e2128954667/content> (дата звернення: 11.05.2025).
 14. Балик А. Тренди дизайну мобільних додатків 2024. URL: <https://lemon.school/blog/trendy-dyzajnu-mobilnyh-dodatkov-2024> (дата звернення: 12.05.2025).
 15. Морозов О. Ю. Біометрична аутентифікація користувачів для захисту веб-додатків: пояснювальна записка до атестаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 125 – Кібербезпека /

- О. Ю. Морозов ; М-во освіти та науки України, Харківський національний університет радіоелектроніки. Харків, 2021. 74 с. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/12e71584-d139-484e-9959-613721cf2a1d/content> (дата звернення: 11.05.2025).
16. Xperience. Вибір кольору у дизайні. URL: <https://ux.pub/uxp/vibir-koloru-u-dizaini-1hl4> (дата звернення: 11.05.2025).
17. Прокопович Т. А. Основи кольорознавства: навчальний посібник. 2-е вид., допов. і перероб. – Луцьк: Волинський національний університет ім. Лесі Українки, 2022. 124 с. URL: https://evnuir.vnu.edu.ua/bitstream/123456789/20686/1/Prokopovych_posibnik_Osnovi_koloroznavstva.pdf (дата звернення: 11.05.2025).
18. Сідельнікова Д. Етапи створення дизайну мобільного застосунку: дисертація на здобуття ступеня доктора філософії. – Київ: Національний авіаційний університет, 2022. URL: <https://er.nau.edu.ua/server/api/core/bitstreams/d91ed23b-d736-4e82-b747-7eb819b04cb6/content> (дата звернення: 11.05.2025).
19. Khoroshevska I. O., Aliksieieva I. D. Creating a Methodology for Developing a Mobile Application Design for Emotional Stabilization // Scientific Notes of Taurida National V.I. Vernadsky University. Series: Technical Sciences. – 2023. – No. 5. – P. 266–272. – DOI: <https://doi.org/10.32782/2663-5941/2023.5/41> (date of access: 11.05.2025).
20. Sandesara M., Bodkhe U., Tanwar S., Alshehri M. D., Sharma R., Neagu B.-C., Grigoras G., Raboaca M. S. Design and Experience of Mobile Applications: A Pilot Survey // Mathematics. – 2022. – Vol. 10, No. 14. – Article 2380. – DOI: <https://doi.org/10.3390/math10142380> (дата звернення: 12.05.2025).
21. Суліма В. К., Зелінська О. В. Сучасні тенденції Frontend мобільних додатків // Прикладні інформаційні технології: збірник тез доповідей. – Вінниця: Донецький національний університет імені Василя Стуса, 2024. URL: <https://jait.donnu.edu.ua/index.php/jait/article/view/17011> (дата звернення: 11.05.2025).

22. Пещерьова В. С. Адаптивний інтерфейс навчальної платформи на основі технологій UX/UI: магістерська атестаційна робота. – Харків: Харківський національний університет радіоелектроніки, 2023. URL: <https://krs.chmnu.edu.ua/jspui/bitstream/123456789/2947/pdf> (дата звернення: 11.05.2025).
23. Воронцова Д. В., Яценко Т. М. Інтерактивне меню у телефонних застосунках: магістерська атестаційна робота. – Харків: Харківський національний університет радіоелектроніки, 2024. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/a164e7d8-947a-4501-8578-c5834fd8d42a/content> (дата звернення: 11.05.2025).
24. Ichanska N., Ulko S. Основні аспекти створення мобільних додатків та вибір інструментів їх розробки // Системи управління, навігації та зв'язку: збірник наукових праць. – 2020. – Т. 1, № 59. – С. 74–78. – DOI: <https://doi.org/10.26906/sunz.2020.1.074> (дата звернення: 11.05.2025).
25. Базилевський О. А. Розробка мобільного застосунку для пошуку телефону за допомогою React Native: кваліфікаційна робота бакалавра. – Запоріжжя: Запорізький національний університет, факультет математичний, кафедра програмної інженерії, 2024. 45 с. URL: https://dspace.znu.edu.ua/xmlui/bitstream/handle/12345/20083/Базилевський_диплом.pdf?sequence=1&isAllowed=y (дата звернення: 11.05.2025).
26. Кацалап О. С. Розробка мобільного додатку для продажу кави та чаю з використанням технології React Native: кваліфікаційна робота бакалавра. – Кривий Ріг: Криворізький національний університет, факультет інформаційних технологій, кафедра автоматизації, комп'ютерних наук і технологій, 2024. 75 с. URL: <https://dspace.knu.edu.ua/handle/123456789/20083> (дата звернення: 11.05.2025).
27. Green B., Seshadri S. AngularJS: Up and Running: Enhanced Productivity with Structured Web Apps. – Sebastopol : O'Reilly Media, 2014. – 319 с. – ISBN 978-1491901946.

28. Карабаш В. Є. Розробка програмного забезпечення для мобільних платформ: кваліфікаційна робота бакалавра. – Полтава: Полтавський університет економіки і торгівлі, 2024. URL: http://dspace.puet.edu.ua/bitstream/123456789/13973/1/Karabash_Vadym_Yevhenovich_KNb41-1.pdf (дата звернення: 11.05.2025).
29. React Native. Accessibility. URL: <https://reactnative.dev/docs/accessibility> (дата звернення: 11.05.2025).
30. React Native. Testing Overview. URL: <https://reactnative.dev/docs/testing-overview> (дата звернення: 11.05.2025).
31. Nguyen N. Improving the Accessibility of REGames Mobile Application in React Native: Bachelor's Thesis. – Helsinki: Metropolia University of Applied Sciences, 2021. URL: <https://www.theseus.fi/bitstream/handle/10024/752536/Bachelor%27s%20thesis%20Elina%20Niemela.pdf?sequence=2&isAllowed=y> (дата звернення: 11.05.2025).
32. Králusz T. A. Mobile Application Development with React Native and Leveraging Third-Party Libraries: дипломна робота. – LUT University, 2024. URL: https://lutpub.lut.fi/bitstream/handle/10024/160734/diplomityö_salohonka_matias.pdf?sequence=1&isAllowed=y (дата звернення: 11.05.2025).
33. Rashid H. Front End Development and UX Design. – Turin: Politecnico di Torino, 2024. URL: <https://webthesis.biblio.polito.it/secure/31821/1/tesi.pdf> (дата звернення: 11.05.2025).
34. Veeri V. Performance Optimization Techniques in React Applications: A Comprehensive Analysis // International Journal of Research in Computer Applications & Information Technology. 2024. Т. 7, № 2. URL: https://www.researchgate.net/profile/Veeranjaneyulu-Veeri/publication/385785715_Performance_Optimization_Techniques_in_React_Applications_A_Comprehensive_Analysis/ (дата звернення: 11.05.2025).

35. Hasan A. Master Lazy Loading Concept in React Native/ReactJS [Електронний ресурс] // DEV Community. URL: https://dev.to/ajmal_hasan/master-lazy-loading-concept-in-react-native-reactjs-3cfe (дата звернення: 11.05.2025).
36. Glos J. Optimizing React Native Application Deployment Processes with Automation: дипломна робота. – Brno: Masaryk University, 2024. URL: <https://is.muni.cz/th/lobsg/thesis.pdf> (дата звернення: 12.05.2025).
37. Дичко С. А. Кросплатформений застосунок «Трекер звичок» для мобільних девайсів: кваліфікаційна робота бакалавра. – Миколаїв: Чорноморський національний університет імені Петра Могили, 2023. URL: <https://krs.chmnu.edu.ua/jspui/bitstream/123456789/2943/1/Дичко%20Сергій%20ОБКР.pdf> (дата звернення: 12.05.2025).
38. Шпортько О. В., Малаш К. М. Застосування мемоізації в задачах мінімізації кількості знаків арифметичних операцій // Вісник Національного університету водного господарства та природокористування. Технічні науки. 2020. Вип. 2. С. 93–101. URL: <https://cyberleninka.ru/article/n/metod-pidvischennya-produktivnosti-vebdodatku-v-ekosistemi-freymvorku-react/viewer> (дата звернення: 11.05.2025).
39. Пироженко С., Лесная Н. Дослідження продуктивності методів та засобів контролю глобального стану компонентів React додатків. 2020. URL: <https://nauka-online.com/wp-content/uploads/2020/10/Pirozhenko-Lyesna.pdf> (дата звернення: 11.05.2025).
40. Ананченко В., Лотюк Ю. Управління React-застосунком через глобальні та локальні стани з використанням сучасних підходів та інструментів // Збірник матеріалів конференції. 2024. С. 147–152. URL: <http://dspace.megu.edu.ua:8080/jspui/bitstream/123456789/5203/1/збірник%20конференції%20частина%201%202024-147-152.pdf> (дата звернення: 11.05.2025).
41. Гуменюк К.В., Січко Т.В. Передові методи анімації вебінтерфейсів: вплив на користувацький досвід. Матеріали V Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених (24 травня 2024 року, м.

Вінниця). Вінниця, ДонНУ імені Василя Стуса, 2024. С. 13-15. URL: <https://jait.donnu.edu.ua/article/view/16981> (дата звернення: 18.05.2025).



ДЕКЛАРАЦІЯ

про дотримання академічної доброчесності

Я, _____

Повністю вказується ПІБ та статус (посада для працівників, освітня (освітньо-наукова) програма – для здобувачів вищої освіти)

що нижче підписалась/підписався, розуміючи та підтримуючи загально визнані засади справедливості, доброчесності та законності,

ЗОБОВ'ЯЗУЮСЬ:

дотримуватися принципів та правил академічної доброчесності, що визначені законодавством України, локальними нормативними актами Донецького національного університету імені Василя Стуса, положеннями, правилами, умовами, визначеними іншими суб'єктами, та не допускати їх порушення.

ПІДТВЕРДЖУЮ:

що мені відомі положення статті 42 Закону України «Про освіту»; що у даній роботі не представляла/представляв чийсь роботи повністю або частково як свої власні. Там, де я скористалася/скористався працею інших, я зробила/зробив відповідні посилання на джерела інформації; що дана робота не передавалася іншим особам і подається вперше, не порушує авторських та суміжних прав закріплених статтями 21-25 Закону України «Про авторське право та суміжні права», а дані та інформація не отримувались в недозволений спосіб.

УСВІДОМЛЮЮ:

що ця робота може бути перевірена університетом на плагіат або інші порушення академічної доброчесності, в тому числі з використанням спеціалізованих сервісів; що у разі порушення академічної доброчесності, до мене можуть бути застосовані процедури, передбачені законодавством України та Кодексом академічної доброчесності та корпоративної етики Донецького національного університету імені Василя Стуса, іншими локальними нормативними актами університету, та я можу бути притягнута/притягнутий до академічної відповідальності.

(дата)

(підпис)